

---

# Aplicación móvil Android para cuadernos de viaje turísticos sincronizada con plataforma web

---

Autor: Sergio Martínez Rodríguez  
Director: José Cascán Fernández  
Ponente: Pedro Muro Medrano



**Dep. de Informática e  
Ingeniería de Sistemas (DIIS)**  
Centro Politécnico Superior  
María de Luna, 1  
50018 Zaragoza (España)



**GeoSpatiumLab S.L.**  
Carlos Marx, 4 Local Izdo  
50015 Zaragoza (España)



**Grupo de Sistemas de  
Información Avanzados (IAAA)**  
Centro Politécnico Superior  
María de Luna, 1  
50018 Zaragoza (España)





*Agradecimientos:*

*A IAAA, por darme la oportunidad de desarrollar este proyecto y brindarme todo lo necesario, así como un pequeño hueco en el grupo, que ha hecho posible la realización del mismo en un ambiente idóneo para ello. En especial a José y Rodolfo, quienes me han iluminado durante todo este camino.*

*A mis compañeros del IAAA, porque han hecho mucho más llevadero y ameno el tiempo que llevo dentro del grupo.*

*A mi familia, por haberme dado la oportunidad de llegar hasta este punto de mi vida, por su bendita paciencia y por haberme enseñado a no rendirme jamás ante cualquier adversidad.*

*Y a Ester, por haber sido la mejor compañera durante todo este tiempo.*

# Aplicación móvil Android para cuadernos de viaje turísticos sincronizada con plataforma web

## RESUMEN

---

GeoSisTur es una plataforma de promoción turística que permite dinamizar un destino utilizando las tecnologías sociales y la optimización del proceso de gestión de contenidos.

Está pensada para satisfacer todas las necesidades que engloba el descubrimiento de un nuevo destino: conocer el lugar a través de diferentes elementos multimedia informativos que impacten en la consciencia del usuario; presentación de los productos turísticos que se pueden disfrutar (qué hacer, comer y beber, itinerarios, etc); organizar la visita utilizando una agenda de viaje donde anotar la planificación decidida; y, tras el viaje, inclusión de comentarios y opiniones en el área de comunidad para que futuros turistas dispongan de información de otros usuarios que han visitado el lugar con anterioridad.

El trabajo realizado en este Proyecto Fin de Carrera tiene como objetivo principal el desarrollo de una aplicación para dispositivos móviles con sistema operativo Android, dedicada a la gestión y planificación de viajes a partir de lo que hemos llamado cuadernos de viaje, basados en el módulo homónimo de GeoSisTur.

En un cuaderno de viaje, el usuario recopilará información sobre un viaje concreto antes, durante o después del periodo que dure dicho viaje. El usuario será capaz de obtener y gestionar desde recursos multimedia, como fotos y videos, hasta recursos turísticos o POI's que ha visitado.

Se ha conseguido desarrollar la aplicación móvil evitando la dependencia total de una conexión a Internet con el fin de conseguir la máxima funcionalidad en cualquier momento y lugar en el tiempo.

Por último, estando la aplicación móvil respaldada por la plataforma web turística GeoSisTur permite el intercambio de contenido entre la plataforma web y el dispositivo móvil.

Las tareas necesarias para la obtención de dicha aplicación móvil han sido las tradicionales en un proyecto software, es decir, la determinación de los requisitos que deberán cumplirse, el análisis y el diseño de la propia aplicación, la implementación de las soluciones diseñadas, y la integración final de los componentes.



# Índice general

<b>I</b>	<b>Memoria</b>	<b>1</b>
<b>1.</b>	<b>Introducción</b>	<b>2</b>
1.1.	Contexto profesional . . . . .	2
1.2.	Contexto tecnológico . . . . .	3
1.2.1.	Android . . . . .	3
1.2.2.	XML . . . . .	5
1.2.3.	SQLite y ORMLite . . . . .	6
1.2.4.	Servicios Web . . . . .	6
1.2.5.	JSON y GSON . . . . .	6
1.3.	Motivación y objetivos del proyecto . . . . .	7
1.4.	Estructura del documento . . . . .	9
<b>2.</b>	<b>Trabajo realizado: Travel Blog</b>	<b>11</b>
2.1.	Situación previa y estudio de mercado . . . . .	11
2.1.1.	Tourist Eye . . . . .	12
2.1.2.	Trip Journal . . . . .	12
2.2.	Análisis de la aplicación . . . . .	12
2.2.1.	Definición de cuaderno de viaje . . . . .	13
2.2.2.	Determinación de requisitos . . . . .	13
2.3.	Diseño de la aplicación . . . . .	14
2.3.1.	Arquitectura del sistema . . . . .	14
2.3.2.	Modelo de datos . . . . .	16
2.4.	Metodología de creación de una aplicación móvil Android . . . . .	17
2.4.1.	Diseño gráfico de la ventana . . . . .	18
2.4.2.	Análisis, obtención y muestra de la información . . . . .	20
2.4.3.	Dotación de interacción . . . . .	23
2.4.4.	Pruebas de integración en la aplicación . . . . .	26
2.5.	Decisiones de diseño adoptadas . . . . .	27
2.5.1.	Representación de la información . . . . .	27
2.5.2.	Calendario . . . . .	28
2.5.3.	Visor de mapas . . . . .	30
2.5.4.	Elementos multimedia . . . . .	32
2.5.5.	Comunicación con GeoSisTur . . . . .	33

<b>3. Conclusiones</b>	<b>35</b>
3.1. Resultados Obtenidos . . . . .	35
3.2. Líneas Futuras . . . . .	35
3.3. Valoración Personal . . . . .	36
 <b>II Anexos</b>	 <b>38</b>
<b>A. Gestión del proyecto</b>	<b>39</b>
A.1. Metodología de trabajo . . . . .	39
A.2. Planificación . . . . .	40
A.3. Herramientas utilizadas . . . . .	41
<b>B. Análisis de Requisitos</b>	<b>43</b>
B.1. Requisitos funcionales . . . . .	43
B.2. Requisitos no funcionales . . . . .	45
<b>C. Diseño y Modelo de datos</b>	<b>46</b>
C.1. Diseño de la aplicación . . . . .	46
C.1.1. Prototipado de las ventanas de navegación . . . . .	46
C.1.2. Navegación de la aplicación . . . . .	47
C.2. Modelo de datos . . . . .	47
<b>D. Arquitectura del sistema</b>	<b>50</b>
<b>E. Manual de instrucciones Travel Blog</b>	<b>53</b>
 <b>III Acrónimos, Figuras, Bibliografía</b>	 <b>76</b>
<b>Acrónimos</b>	<b>77</b>
<b>Índice de figuras</b>	<b>78</b>
<b>Bibliografía</b>	<b>79</b>

Parte I

Memoria

# Capítulo 1

## Introducción

El presente documento tiene como objetivo recoger toda la información relacionada con la realización de este Proyecto Fin de Carrera (**PFC**) titulado *Aplicación móvil Android para cuadernos de viaje turísticos sincronizada con plataforma web*. En este primer capítulo se define el contexto, tanto profesional como tecnológico, en el que se ha desarrollado dicho proyecto. A continuación se exponen los objetivos planteados al principio del proyecto y la estructura en la que se organiza este documento.

### 1.1. Contexto profesional

Este PFC se ha llevado a cabo en GeoSpatiumLab S.L.<sup>1</sup>, empresa especializada en el desarrollo de Sistemas de Información en los que la componente geográfica resulta fundamental. GeoSpatiumLab surge como Spin-Off de la Universidad de Zaragoza, entre cuyos objetivos está facilitar la transferencia de tecnología generada por el Grupo de Sistemas de Información Avanzados (IAAA), perteneciente al Departamento de Informática e Ingeniería de Sistemas. El Grupo de Sistemas de Información Avanzados es un grupo de **I+D+i**<sup>2</sup> de carácter multidisciplinar, pero con un marcado perfil informático. Está adscrito al Instituto de Investigación en Ingeniería de Aragón (I3A) de la Universidad de Zaragoza, y además es Grupo de Investigación Consolidado reconocido por la **DGA**<sup>3</sup> (código T56). La actividad de investigación de este grupo está enfocada en las tecnologías de software abierto, distribuido e interoperable, para la creación de sistemas de información con datos georreferenciados (también denominados generalmente datos geográficos y ahora más comúnmente datos espaciales). El presente proyecto surge como una oportunidad de aplicar la tecnología desarrollada por las diferentes líneas de investigación del Grupo en un producto novedoso destinado a un sector en constante expansión como

---

<sup>1</sup><http://www.geospatiumlab.es/>

<sup>2</sup>Investigación, desarrollo e innovación

<sup>3</sup>Diputación General de Aragón

es el turístico, y usando una tecnología en auge como es el desarrollo de aplicaciones móviles.

## 1.2. Contexto tecnológico

Antes de profundizar en las cuestiones técnicas del proyecto, y para una mejor comprensión del desarrollo del mismo, es preciso presentar una breve descripción de las tecnologías y herramientas más importantes que han tenido cabida en la realización de este PFC.

### 1.2.1. Android

Android<sup>4</sup> es un sistema operativo para dispositivos móviles tales como teléfonos inteligentes o tabletas. Está desarrollado por la Open Handset Alliance, una alianza comercial de 78 compañías tan importantes como Samsung, HTC, Dell, Intel, etc, para desarrollar estándares abiertos para dispositivos móviles, lideradas por Google.

El sistema operativo Android se compone de aplicaciones que se ejecutan en un framework Java de aplicaciones orientadas a objetos, sobre el núcleo de las bibliotecas de Java en una máquina virtual Dalvik con compilación en tiempo de ejecución. Este sistema operativo contiene una base de datos relacional SQLite, de la que hablaremos más adelante.

Java es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de los años 90, y se trata del lenguaje utilizado para el desarrollo de aplicaciones para la plataforma Android.

Además, Java es el lenguaje más popular en la comunidad de desarrolladores, por ello es una de las razones por las que el desarrollo de aplicaciones Android se ha puesto de moda en los últimos años.

En Android, las aplicaciones desarrolladas para una versión concreta del sistema operativo son compatibles para dicha versión y para las posteriores. Es decir, a menor versión de la aplicación desarrollada, mayor será el número de dispositivos móviles con los que será compatible. La versión Android para móviles más avanzada al inicio del proyecto era la 2.3 (Gingerbread), pero como he comentado antes, el objetivo es llegar al máximo de dispositivos móviles, por lo que la versión elegida para el desarrollo del PFC es la 2.2 (Froyo). En la figura 1.1 se puede observar el porcentaje de uso para cada una de las versiones existentes en el momento del inicio del proyecto.

Android ofrece una plataforma abierta de desarrollo, lo que proporciona a los programadores la capacidad de construir aplicaciones realmente interesantes e innovadoras.

---

<sup>4</sup><http://www.android.com/>

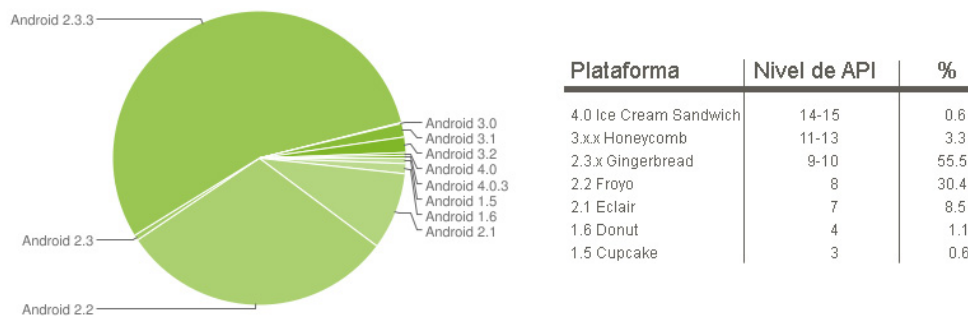


Figura 1.1: Reparto de las versiones de Android a fecha de 3 de enero de 2012

Se podría decir que tras una aplicación Android existe un conjunto de servicios y sistemas:

- Un extenso conjunto de Views u objetos visuales, que son usadas para contruir la interfaz gráfica de una aplicación, incluyendo listas, tablas, botones, etc.
- Content Providers o proveedores de contenido, que permiten el intercambio de información entre aplicaciones.
- Un gestor de recursos, ofreciendo acceso directo a archivos utilizados en una aplicación como imágenes, frases predefinidas o documentos de diseño (layouts).
- Un gestor de notificaciones, que proporciona a toda aplicación la visualización de alertas personalizadas en la barra de estado de la pantalla.
- Un gestor de Activities o actividades, que gestiona el ciclo de vida de una aplicación y proporciona una navegación común.

Una aplicación Android está formada por una serie de componentes esenciales. Cada componente es un punto de unión entre el sistema operativo y la aplicación en sí.

Para este proyecto, únicamente vamos a comentar algunos de los más importantes:

- Activity (Actividad): Una Activity es un componente de la aplicación que ofrece una pantalla con la cual los usuarios pueden interactuar para realizar alguna tarea, como marcar un teléfono, tomar una foto o navegar por un mapa. En este **PFC**, cada Activity se va a corresponder con cada una de las ventanas de navegación obtenidas en la fase de diseño.

Travel Blog va a estar formada por varias Activities que se invocan e intercambian información entre sí. En un proyecto Android, una Activity se trata de una clase Java, así formando parte del código fuente de la aplicación.

- Content Provider: Un Content Provider almacena y recupera datos y los hace accesibles a todas las aplicaciones. Es el único modo de compartir información entre las aplicaciones instaladas en el dispositivo móvil.

En el desarrollo de nuestra aplicación los Content Provider han tomado parte, por ejemplo, a la hora de comunicarnos con Google Calendar, aplicación nativa instalada en todos los dispositivos Android.

- Intent: Un Intent es el elemento básico de comunicación entre los distintos componentes Android que hemos descrito antes. Se pueden entender como los mensajes que son enviados entre los distintos componentes de una aplicación o entre distintas aplicaciones.

En Travel Blog se van a usar precisamente para navegar entre las ventanas que componen la interfaz y para invocar la ejecución de otras aplicaciones, como la cámara nativa de Android, la Gallery, etc.

Además de todos los componentes anteriores, existe un documento esencial en todo proyecto Android llamado *AndroidManifest*. Se trata de un archivo **XML**<sup>5</sup> encargado de la configuración de la aplicación. Algunas de las principales funciones de dicho archivo se enumeran a continuación:

- Identificar los permisos de usuario que la aplicación requiere, tales como acceso a Internet, o a la cámara de fotos, etc.
- Declarar la versión de la **API**<sup>6</sup> de Android que se va a utilizar. En nuestro proyecto la aplicación parte de la version 2.2, lo que equivale a la **API** 8.
- Declarar las características del software y hardware utilizado o requerido por la aplicación, tales como pantalla multitouch o servicio de **GPS**<sup>7</sup>.
- Definir la configuración de cada una de las Activities que van a formar parte de la aplicación como el uso de sensor de giro, o la aparición de la entrada de texto automática.

### 1.2.2. XML

**XML**, es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C)<sup>8</sup>. **XML** no sólo ha nacido para su aplicación en Internet. Concretamente, en el proyecto tiene el papel de definir el diseño gráfico de las distintas pantallas de las que consta la aplicación móvil.

**XML** también tiene un papel muy importante en el archivo de configuración, *AndroidManifest.xml*, presente en todo proyecto Android, además es el clásico estándar

---

<sup>5</sup>eXtensible Markup Language

<sup>6</sup>Application Programming Interface

<sup>7</sup>Global Positioning System

<sup>8</sup><http://www.w3c.es/>

para el intercambio de datos, por ejemplo, con un servicio web, aunque como veremos más adelante, se optará por otro estándar, JSON (JavaScript Object Notation).

### 1.2.3. SQLite y ORMLite

SQLite es un motor de bases de datos **SQL**<sup>9</sup>, contenido en una relativamente pequeña ( 275 KB) biblioteca en C. Por tratarse de un motor de base de datos liviano, Android la adoptó para usarla como almacenamiento de datos en su sistema operativo.

ORMLite se trata de un framework de código abierto que proporciona un mapeo objeto-relacional (**ORM**<sup>10</sup>) simple entre clases Java y bases de datos **SQL**.

Siendo que SQLite se trata de una base de datos **SQL**, ORMLite nos facilitará la comunicación entre la aplicación Android y el almacén de datos del móvil.

### 1.2.4. Servicios Web

Un servicio web, es una pieza de software que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Es el modo elegido en el presente **PFC** para la comunicación entre la aplicación móvil y el portal web GeoSisTur[1]<sup>11</sup>. Además, también se usará para satisfacer otras funcionalidades de la aplicación, como la búsqueda por texto de un lugar en el mapa.

Existen varios estándares de comunicación con servicios web, tales como **WSDL**<sup>12</sup> y **SOAP**<sup>13</sup>, pero para este **PFC** se ha elegido el estándar **REST**<sup>14</sup>, que no es otro que el establecimiento de una comunicación sencilla a través de comandos **HTTP**<sup>15</sup> como POST, GET, PUT o DELETE.

### 1.2.5. JSON y GSON

**XML** ha sido el estándar para el intercambio de datos entre distintas aplicaciones, pero desde hace un tiempo, su uso se está viendo desplazado por otro estándar de mensajes más popular, ligero y fácil de interpretar como es JSON (JavaScript Object Notation).

Además, el uso de **JSON**<sup>16</sup> se hace todavía más útil para este **PFC** usando Google-GSON. GSON se trata de una librería Java de código abierto que nos ayuda a la

---

<sup>9</sup>Structured Query Language

<sup>10</sup>Object-Relational Mapping

<sup>11</sup><http://www.geosistur.com/>

<sup>12</sup>Web Services Description Language

<sup>13</sup>Simple Object Access Protocol

<sup>14</sup>Representational State Transfer

<sup>15</sup>HyperText Transfer Protocol

<sup>16</sup>JavaScript Object Notation



serialización de objetos Java a **JSON**, y viceversa. Es decir, a partir de un objeto Java puede crear un mensaje JSON, y al revés, a partir de un mensaje JSON puede obtener un objeto Java. Nos será de gran ayuda a la hora de intercambiar información con GeoSistur.

### 1.3. Motivación y objetivos del proyecto

Hoy en día, smartphones y tablets están al alcance de cualquiera. Poco a poco están desplazando el uso de ordenadores convencionales y eso ha hecho que se haya convertido en un mercado realmente llamativo para cualquier empresa dedicada a las Tecnologías de la Información y la Comunicación (TIC). El desarrollar aplicaciones para plataformas móviles, como Android, hoy en día es una línea de negocio que muchas empresas, incluso desarrolladores por cuenta propia, están teniendo muy presente.

Para 2012 se espera el boom en el desarrollo de aplicaciones para Android, cada vez más especializadas y gráficamente más atractivas, e incluso se piensa que Android podría llegar este mismo año a otras plataformas como la televisión. Es tal el auge de esta corriente tecnológica, que es difícil dejar pasar la oportunidad.

Por otro lado, en el ámbito turístico, todavía existe un vacío en lo que a aplicaciones móviles se refiere. Muchas ciudades o zonas turísticas tienen su propia aplicación móvil para ofrecer al turista, mostrándole información sobre el lugar en cuestión. Pero la mayoría de ellas no dan al usuario la capacidad de crear y compartir información personal, como experiencias o consejos, sobre ese lugar, algo actualmente muy importante, y más teniendo en cuenta el éxito de las redes sociales. Existe un pequeño número de aplicaciones que cumplen esta característica y pueden considerarse importantes en el área de la gestión y planificación de viajes. Precisamente es ésta el área donde quiere introducirse la aplicación móvil Android a desarrollar a lo largo este **PFC**.

Además, se parte de que en la empresa ya existe una plataforma web turística con soporte a comunidad de usuarios, con gestión y planificación de viajes operativa, pudiendo respaldar a la aplicación móvil. El nombre de esta plataforma es GeoSisTur. La unión de las dos plataformas, móvil y web, genera un producto de gran interés, tanto para el usuario de a pie como para lugares y zonas turísticas que quieran darse a conocer a través de la plataforma.

El propósito de este **PFC** es el desarrollo de una aplicación para dispositivos móviles sobre plataforma Android, que facilite al usuario la creación, consulta y modificación de cuadernos de viaje y su posterior sincronización en la cuenta del usuario dentro de la plataforma web turística GeoSisTur.

Un cuaderno de viaje consiste en el diario de un viaje de un usuario, comentando y valorando los recursos turísticos de la plataforma Web visitados, con la posibilidad de aportar elementos multimedia obtenidos durante el viaje y compartir dicho cuaderno con el resto de la comunidad social.

El cuaderno se podrá planificar antes de la realización del viaje, de manera que la aplicación podrá ser utilizada como guía de viaje, al disponer de la opción de guardar las visitas a los recursos turísticos como eventos en la agenda integrada en el dispositivo móvil. En la aplicación a desarrollar, este tipo de eventos tiene la opción de guardarse en Google Calendar, agenda preinstalada en todo dispositivo Android.

Durante el viaje, el usuario podrá conocer los diferentes recursos turísticos que se encuentren en las proximidades activando los recursos de ubicación espacial del dispositivo, obtener información más detallada de los mismos, reproducir recursos multimedia geolocalizados, y asociar estos recursos a sus cuadernos de viaje si no lo están ya.

La aplicación será funcional tanto con conexión a Internet(online) como sin ella (offline), de manera que se dispondrá de un mapa en todo momento donde poder consultar la posición, no sólo de los recursos turísticos citados anteriormente, sino también de los elementos creados por el usuario que estén geolocalizados, tales como las fotografías obtenidas sobre un recurso turístico visitado que cuenten con información de su posición.

Además se permite la sincronización de los datos del usuario con su cuenta en la plataforma turística.

Esta sincronización se realizará mediante la utilización de servicios Web disponibles en la plataforma, y tendrá efecto en ambos sentidos, actualizando en el dispositivo los elementos del usuario creados en la plataforma, y en la plataforma los elementos aportados mediante la aplicación móvil.

Más detalladamente, el proyecto constará de las siguientes tareas:

- Realizar el estudio previo de la aplicación móvil a desarrollar mediante el análisis de la misma y la búsqueda de requisitos, tanto funcionales como no funcionales, que permitan tener una idea clara de lo que va a ser la funcionalidad en conjunto de la aplicación. Al mismo tiempo, analizar y entender el módulo de los cuadernos de viaje implementado en GeoSisTur, para que la búsqueda de requisitos resulte más sencilla.
- Estudio de otras aplicaciones del mismo ámbito atendiendo a sus funcionalidades para mejorar las prestaciones, experiencia y usabilidad proporcionada al usuario en la aplicación a desarrollar.
- Realizar un diseño inicial y visual de la aplicación Android aglutinando todas las funcionalidades encontradas durante el análisis de la misma.
- Mediante formación, entender y conocer todo el entorno que proporciona Android al programador para tener en cuenta las posibilidades gráficas y de desarrollo de las distintas funcionalidades obtenidas en las fases de análisis y diseño.

- Establecer un modelo de datos que permita segmentar eficientemente la información a almacenar creada por la aplicación cuando no se encuentre conectada a Internet (offline).
- Estudio de las distintas opciones de representar mapas en la plataforma Android y elección de la más adecuada al proyecto.
- Creación de un servicio web que mantendrá la comunicación y sincronización de los datos entre la aplicación móvil y el portal web GeoSistur. Se discutirán las distintas opciones de servicios web y se darán razones que defiendan la opción elegida.

## 1.4. Estructura del documento

El presente documento está dividido en tres partes diferenciadas. La primera parte, dedicada a la Memoria en sí del PFC, y a la que pertenece este capítulo de introducción, pretende aportar una visión global del proyecto sin entrar en detalles de bajo nivel, y presentar las conclusiones a las que se ha llegado mediante la realización del mismo. La segunda parte está comprendida por anexos que detallan con mayor precisión diferentes aspectos de la labor realizada, del proceso llevado a cabo y de los resultados obtenidos. Por último, se incluye una tercera parte conteniendo un listado de los acrónimos utilizados, el índice de las figuras que aparecen en el texto y las referencias bibliográficas consultadas.

### ■ Parte I. Memoria

- **Capítulo 1. Introducción.** Capítulo al que pertenece esta sección, en el que se presentan el contexto del Proyecto Fin de Carrera objeto de este documento y su finalidad.
- **Capítulo 2. Trabajo Realizado: Travel Blog.** En este capítulo se detallan las labores de análisis, diseño e implementación de la aplicación móvil, a la que se ha llamado Travel Blog, así como los hitos conseguidos durante la realización de la misma.
- **Capítulo 3. Conclusiones.** Capítulo dedicado a las conclusiones obtenidas con la realización del PFC, tanto de carácter profesional como personal. Además se evaluará el cumplimiento de los objetivos y se expondrán las líneas futuras de trabajo que pueden seguirse gracias a los resultados logrados.

### ■ Parte II. Anexos

- **Anexo A. Gestión del proyecto.** Anexo donde se presentan la metodología y la planificación seguidas durante la realización del proyecto, así como un breve resumen de las herramientas utilizadas.

- **Anexo B. Análisis de requisitos.** En este anexo se detalla en profundidad el conjunto de requisitos que se determinaron durante las labores de análisis.
  - **Anexo C. Diseño y modelo de datos.** En este anexo se detallan la fase de diseño y las decisiones en la organización de las entidades que forman el modelo de datos utilizado en el proyecto.
  - **Anexo D. Arquitectura del sistema.** En este anexo se recoge la arquitectura del sistema pensada para Travel Blog en detalle.
  - **Anexo E. Manual de usuario de aplicación móvil Travel Blog.** Una de las labores realizadas para la aplicación móvil desarrollada fue la confección de un manual de usuario. En este anexo se recoge dicho manual.
- **Parte III. Acrónimos, Figuras y Bibliografía**
- **Acrónimos**
  - **Índice de figuras**
  - **Bibliografía**

## Capítulo 2

# Trabajo realizado: Travel Blog

Este capítulo está dedicado a detallar todo el trabajo realizado a lo largo de este **PFC**. El nombre que se ha elegido para llamar a la aplicación que ha sido desarrollada durante el periodo que ha durado el **PFC** y a la cual se va a hacer referencia ha sido Travel Blog.

A continuación, se muestra la situación previa desde donde parte el proyecto, así como un estudio sobre algunas aplicaciones ubicadas en el mismo ámbito que Travel Blog. Más tarde se explican cada una de las partes de las que consta la realización de un proyecto software, análisis, diseño e implementación, incluyendo la metodología que se ha aplicado para llegar a la elaboración de Travel Blog.

### 2.1. Situación previa y estudio de mercado

En la empresa donde se ha desarrollado este **PFC** se cuenta con una plataforma web dedicada al turismo llamada GeoSisTur. Actualmente, una de las corrientes que existen en la actualidad tecnológica pretende llevar todo el contenido disponible en la web a la plataforma móvil.

El módulo de cuadernos de viaje de GeoSisTur es uno de los contenidos que se desea importar al móvil, por ello se busca desarrollar una aplicación móvil, en principio para Android, que cumpla muchas de las funcionalidades del módulo, y además exista una comunicación entre la aplicación y la plataforma web.

Como uno de los puntos fuertes del proyecto, y debido en gran parte a la poca disponibilidad de acceso a Internet mientras se viaja, ya sea por falta de cobertura o por tarifas desorbitadas, se pensó en que la aplicación tuviera la máxima funcionalidad en modo offline.

Se buscaba crear una aplicación que disponga de unas cualidades de aspecto innovador y que fueran capaces de diferenciar la aplicación Travel Blog con otras ya existentes en el mercado. Precisamente por eso se analizaron también otras aplicaciones del mismo ámbito, la gestión y planificación de viajes por parte del usuario.

### 2.1.1. Tourist Eye

Tourist Eye<sup>1</sup> se trata de una aplicación móvil, que al igual que la que se desarrolla en este PFC, parte de complemento a un gran portal web turístico. El portal web es muy completo, y ofrece una gran cantidad de ventajas de cara al usuario para planificar su viaje. Aún así, usando la aplicación para Android, se puede ver que no han sabido llevar de una manera correcta el sistema creado en la web al móvil. Como ventajas, comentar que la aplicación da gran cantidad de información al usuario, y la interfaz es simple y sencilla. De otro modo, da la sensación de que el usuario dispone de poca libertad a la hora de crear lugares propios a compartir con el resto de la comunidad. Además, carece de un mapa global, algo que en la aplicación móvil a desarrollar se le ha dado una gran importancia.

### 2.1.2. Trip Journal

Trip Journal<sup>2</sup> es una aplicación muy premiada en el área turística y que goza de gran prestigio. Visualmente es muy atractiva, con una interfaz muy cuidada y una navegabilidad agradable. Ofrece muchas de las funcionalidades definidas para este PFC.

La gran diferencia radica en el modo de sincronización con Internet. Trip Journal carece de una plataforma web turística a modo de comunidad que la respalde, como son GeoSisTur o Tourist Eye, sin embargo, centra su atención en compartir la información creada por el usuario a través de otras plataformas sociales como Facebook, Picasa, Google Earth, etc. En la aplicación a desarrollar por el PFC, no se contempla este caso, pero es una idea muy atractiva para incluir en líneas futuras.

Aparte, mencionar que dicha aplicación está disponible para las plataformas móviles más importantes del momento, Bada, iPhone, Symbian y Android, aunque también tiene el inconveniente de que se trata de una aplicación de pago.

## 2.2. Análisis de la aplicación

En esta sección se presenta la fase de análisis, parte inicial de todo proyecto software, donde se presentan los requisitos, tanto funcionales como no funcionales de la aplicación móvil, así como una vista global de la arquitectura pensada para el sistema.

---

<sup>1</sup><http://www.touristeye.es/>

<sup>2</sup><http://www.trip-journal.com/>

### 2.2.1. Definición de cuaderno de viaje

Partiendo del análisis del módulo de mismo nombre de GeoSisTur, se puede definir un cuaderno de viaje como un diario o bitácora en el que el usuario puede relatar, día a día, sus experiencias vividas durante un viaje. Consta de un periodo acotado de acción, de una descripción (del diario completo) y de un conjunto de entradas para cada uno de los días del viaje, acompañadas de fotos, vídeos, recursos turísticos, etc., estos últimos con sus valoraciones y comentarios pertinentes.

### 2.2.2. Determinación de requisitos

En primer lugar, como en cualquier proyecto software, se procede a determinar los requisitos que deberán satisfacer el sistema. Estos requisitos deberán ser cumplidos en su conjunto por la aplicación móvil, además, la aplicación desarrollada deberá cumplir el papel de aplicación base, sujeta a posibles cambios por parte del cliente.

En este punto hay que resaltar que la aplicación va a ser un producto genérico concebido por la empresa con objeto de ser utilizada junto con GeoSisTur, plataforma dedicada a la construcción de portales turísticos, por lo que en la obtención de los requisitos no ha participado ningún cliente final, sino que se han elaborado según un amplio concepto que se ha formado del producto.

En el anexo **B** se especifican con mayor nivel de detalle los requisitos resumidos a continuación de la siguiente manera.

#### **Requisitos funcionales:**

Estos requisitos definen las acciones fundamentales que debe realizar el software al recibir información, procesarla y producir resultados.

- Funcionamiento offline.
- Gestión de cuadernos de viaje.
- Gestión de recursos turísticos.
- Gestión de recursos multimedia.
- Geolocalización de recursos.
- Localización de recursos en mapa de situación.
- Localización del usuario sobre el mapa.
- Visualización offline de una zona concreta del mapa.
- Visualización de contenido multimedia.
- Comentar y valorar recursos turísticos.

- Conexión con GeoSisTur.
- Búsqueda de contenido.
- Búsqueda de recursos turísticos.
- Búsqueda de lugares geográficos.
- Indicación de ruta desde origen hasta destino.

#### **Requisitos no funcionales:**

Los requisitos no funcionales marcan las características que debe poseer el sistema en cuanto a rendimiento, seguridad, portabilidad, y otros aspectos que no forman parte directamente del funcionamiento (entradas, procesamiento, salidas) del mismo. Estos requisitos son los siguientes:

- Creación de aplicación móvil para la plataforma Android de Google.
- Uso de tecnologías *Open Source* (Código Abierto) en el visor de mapas, concretamente OpenStreetMaps (OSM).
- Gestión y persistencia de objetos Java en base de datos SQLite a través de ORMLite.
- Conexión con servicios web externos mediante protocolo **REST** y formato de mensajes **JSON**.

## **2.3. Diseño de la aplicación**

Una vez definidos los requisitos que deberán cumplirse por la aplicación móvil, la siguiente tarea consiste en plantear el diseño de la solución. En primer lugar se presentará la visualización de la arquitectura del sistema a un nivel alto de abstracción, y a continuación se detallará el modelo de los datos utilizado.

### **2.3.1. Arquitectura del sistema**

El sistema está constituido por tres partes diferenciadas. La primera parte es la propia aplicación en sí, es decir, todo el código Java, clases, archivos de configuración, imágenes, etc, que forman el proyecto de programación. También se incluyen otras aplicaciones ya instaladas en el dispositivo móvil, como Gallery, Google Calendar o Camera.

La segunda parte del sistema son las librerías externas a Android añadidas al proyecto, las cuales aportan nuevas herramientas al entorno de desarrollo.

La tercera parte está constituida por las fuentes de datos en las que se apoya y se nutre la aplicación. Existen dos tipos de fuentes de datos:



- Internas: localizadas la propia memoria interna del dispositivo móvil, como la tarjeta de memoria **SD**<sup>3</sup>, donde se almacenan archivos tales como fotos, vídeos, rutas, etc, así como la base de datos SQLite, sistema de almacenamiento propio de Android y donde guardaremos todos los datos creados desde la aplicación.
- Externas: son todas aquellas fuentes de información ajenas al dispositivo móvil. La mayoría de ellas serán accesibles mediante servicios web, como el que mantendrá la comunicación con GeoSisTur, o como el que realiza la búsqueda de un lugar geográfico a partir de una entrada de texto(Google Geocoding).

A su vez, estas tres partes diferenciadas del sistema están comunicadas entre sí. La comunicación entre la capa de aplicación y la capa de librerías se realiza directamente o a través del *Application Framework* de Android. El *Application Framework* es la zona del sistema operativo donde se encuentran las herramientas que Android proporciona al desarrollador de aplicaciones. Se podría decir que esta zona de la arquitectura se encarga de la conexión entre la aplicación y el exterior. Alguna de las herramientas incluidas en el *Application Framework* se enumeran a continuación:

- Gestor de Activities.
- Sistema de Views.
- Gestor de recursos (iconos, audio, estilos, etc).
- Gestor de localización.
- Gestor de telefonía.

La comunicación entre la capa de aplicación y las fuentes de datos va a depender del tipo de fuente de información. En el caso de las fuentes internas, la capa de aplicación utiliza librerías propias de Android o añadidas al proyecto para establecer la conexión.

En el caso de las fuentes externas, la comunicación se establece mediante la red de datos disponible en el móvil.

Se pueden ver los métodos de comunicación con las diferentes fuentes de datos en la sección **2.4.2**.

Esta breve introducción de la arquitectura del sistema que comprende este proyecto puede entenderse mejor observando la figura **2.1**.

En el anexo **B** se especifica con un mayor nivel de detalle la arquitectura del sistema, además de explicarse cada uno de los módulos de los que consta el sistema.

---

<sup>3</sup>Secure Digital

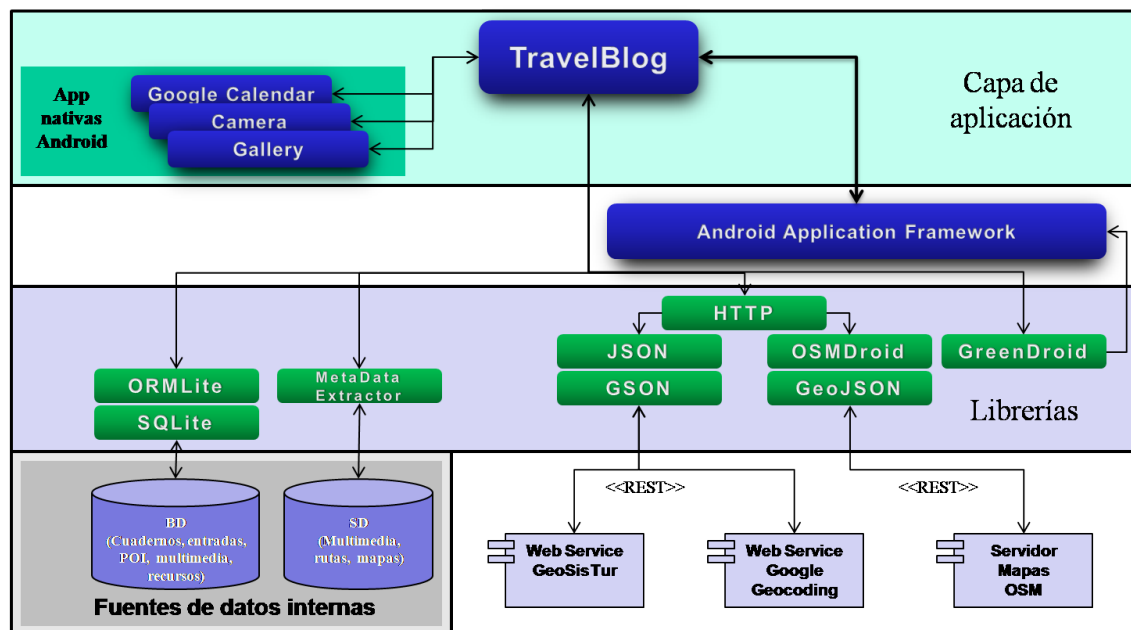


Figura 2.1: Arquitectura del sistema de la aplicación Travel Blog

### 2.3.2. Modelo de datos

Tras la definición de un cuaderno de viaje, la fase de análisis en busca de los requisitos a cumplir por el sistema y el diseño de la arquitectura del sistema, es necesario definir la organización de las entidades que contienen la información. Por ello se define el modelo de datos interno de la aplicación móvil.

Lo primero es obtener cada una de las entidades que van a entrar en juego en el modelo de datos de la aplicación. Más tarde se definen las relaciones existentes entre cada una de las entidades dentro del modelo de datos de la aplicación. Se puede ver el modelo de datos desarrollado entre las entidades obtenidas en la figura 2.2:

Se puede observar que todo el contenido gira alrededor de la entidad entrada (Entry), cada una de las partes que forman un cuaderno de viaje (Travel Blog). Cada entrada aglutina la información recopilada en forma de los diferentes recursos, y forma parte un cuaderno de viaje.

En el anexo C se especifica con un mayor nivel de detalle el modelo de los datos utilizado en el diseño de la aplicación, así como los métodos y atributos pertenecientes a cada entidad.

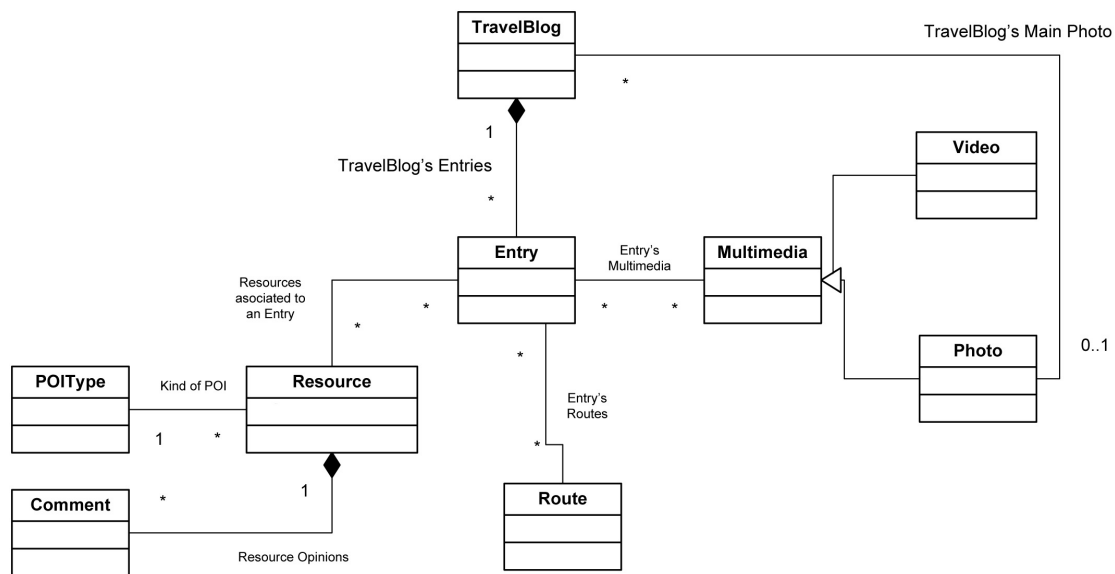


Figura 2.2: Modelo de datos de la aplicación móvil Travel Blog

## 2.4. Metodología de creación de una aplicación móvil Android

El desarrollo de la aplicación móvil se lleva el grueso del trabajo realizado durante el **PFC**.

Antes de entrar en la metodología utilizada para el desarrollo de Travel Blog, mencionar que la fase a la que se relaciona con el diseño gráfico de la aplicación no podía ser realizada sin tener conocimiento alguno de los recursos gráficos disponibles en Android. Por ello, en todo momento, la fase de implementación y de diseño gráfico de la aplicación se han desarrollado a la par. Se iniciaron al mismo tiempo debido a que por la falta de conocimientos en programación de aplicaciones para móvil Android, antes de comenzar a diseñar partes de la aplicación que fueran inviables de llevar a cabo en la fase de desarrollo, se pensó en conocer las distintas opciones de diseño gráfico que Android ofrecía.

Una vez conocido lo básico de los recursos gráficos proporcionados por el **SDK**<sup>4</sup> de Android, y tras diversos tests realizados modularmente con cada uno de ellos, se comenzó a diseñar una aproximación de interfaz gráfica en base a lo que se había probado.

En la figura 2.3 podemos ver alguna de las ventanas principales del diseño gráfico de la aplicación.

Además, contamos con la ventaja de que Android nos permite plasmar un diseño previo durante la fase de desarrollo de la aplicación, ya que cada una de las ventanas de navegación diseñadas se corresponden con una parte de código fuente

<sup>4</sup>Software Development Kit

independiente.

Lo comentado anteriormente y otras muchas cualidades más de la organización de un proyecto Android, hacen que el desarrollo de aplicación se vuelva sencillo y modular. Por ello, la política de desarrollo de la aplicación que se ha seguido, se ha basado en la iteración de la siguiente metodología de programación para cada una de las ventanas de navegación obtenidas en la fase de diseño:

- Diseño gráfico de la ventana mediante la edición del **XML** correspondiente y con la ayuda del asistente del entorno de programación.
- Análisis de la información necesaria a utilizar en dicha ventana de navegación, obtención y muestra de la misma.
- Dotación de interacción con el usuario a cada uno de los elementos gráficos Android (botones, menú, barra de acción, etc).
- Pruebas de los distintos componentes de la aplicación.



Figura 2.3: Ventanas de la aplicación realizadas durante la fase de diseño

### 2.4.1. Diseño gráfico de la ventana

En una aplicación Android, la interfaz de usuario se construye a partir de objetos View o ViewGroup, grupo de varias Views. Estos objetos son las unidades básicas que conforman la interfaz de usuario en la plataforma Android.

La clase View sirve como base para subclases llamadas *widgets* u objetos de la interfaz de usuario completamente implementados, como lo son botones o campos de texto.

Mientras que la clase ViewGroup sirve como base para subclases llamadas *layouts*, que ofrecen diferentes tipos de estructura al diseño de la interfaz. Existen diferentes tipos de *layouts* como lineales, tabulares y relativos.

Un objeto View es un elemento gráfico de Android. Cada View tiene propiedades que almacenan parámetros de diseño y contenido para un área rectangular específica de la pantalla. Algunos de estos parámetros pueden ser las dimensiones del objeto, el diseño, el estilo, posicionamiento respecto a otros Views, etc. Además, cada una de estas View hereda algún que otro parámetro del ViewGroup que lo contiene.

Un objeto View es también un punto de interacción para el usuario y un receptor de eventos de interacción. En la sección 2.4.1, donde se habla de la dotación de la interacción, comentaremos de manera general los eventos táctiles detectables y los correspondientes métodos de activación usados para crear la interacción deseada entre el usuario y la aplicación Android.

El método más común de definir el diseño gráfico y expresar la jerarquía de los elementos que forman la vista de nuestra aplicación es a través de archivos de diseño XML o *layouts*. Este archivo está compuesto por diferentes objetos View o ViewGroup formando una estructura que se puede entender como un árbol, donde las ramas son los objetos ViewGroup, y las hojas los objetos View. Todos en conjunto definen el aspecto gráfico de una interfaz de la aplicación.

En la fase de desarrollo de la aplicación de este PFC, cada uno de los archivos de diseño XML define el diseño gráfico de cada una de las ventanas de navegación obtenidas durante la fase de diseño.

Aparte de definir el diseño de una ventana mediante archivos XML, existe otra manera de hacerlo. Esta manera consiste en definir directamente sobre el código fuente cada uno de los *widgets* a incluir en la interfaz. Es una forma menos intuitiva de diseñar interfaces, pero ofrece al programador una gran ventaja: dinamismo.

Mientras que el archivo XML es un archivo estático, independiente de la ejecución de la aplicación, y que sólo se tiene en cuenta en la ejecución inicial de la ventana, el crear las interfaces directamente dentro del código fuente de la aplicación hace que la interfaz sea dinámica.

El entorno de programación utilizado en este PFC, y que nombramos en el anexo A.3, incluye un editor de interfaz de ventanas en Android para los archivos de diseño XML, el cual nos puede dar una imagen previa del diseño definitivo.

Aunque no sea una herramienta muy cómoda ni tampoco muy usable, sólo por el hecho de poder observar una imagen previa aproximada de la interfaz, nos ahorrará mucho más trabajo que compilando la aplicación y observando la interfaz en el dispositivo móvil. Además, nos permite conocer el aspecto gráfico de todos los distintos *widgets* que incluye el SDK de Android.

En la figura 2.4 podemos ver un ejemplo de implementación de la interfaz gráfica. Se puede observar la parte visual y la parte codificada del archivo de diseño o *layout* durante el desarrollo del mismo.

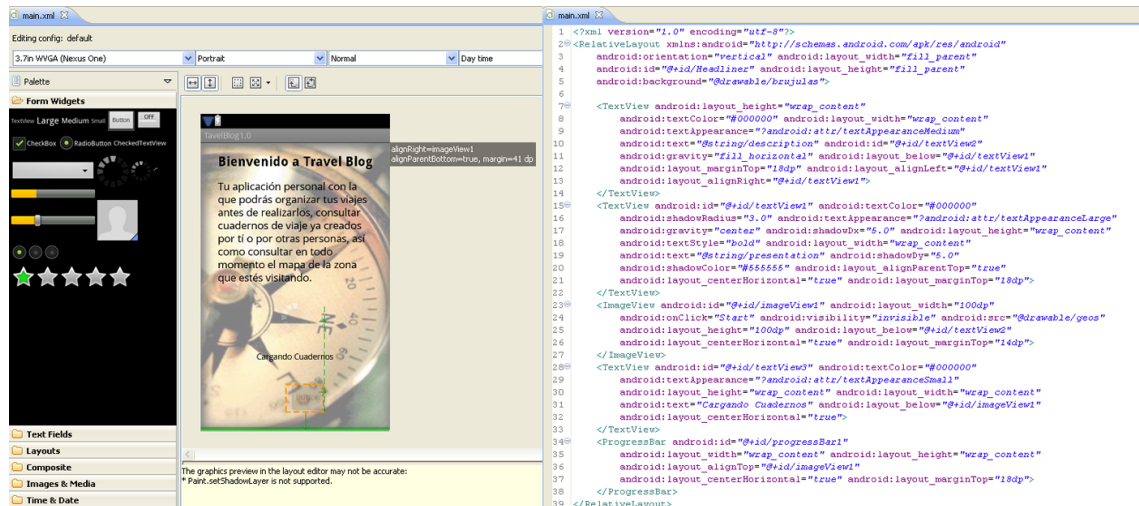


Figura 2.4: Editor XML del entorno de programación utilizado para el diseño de la interfaz de usuario de TravelBlog

### 2.4.2. Análisis, obtención y muestra de la información

Una vez elaborada la interfaz gráfica de una ventana en cuestión y en base al modelo de datos diseñado, se procede a realizar un análisis de la información necesaria que debemos tener, para después darle la funcionalidad deseada a los *widgets* o elementos que componen dicha ventana.

Esta información la obtenemos a partir de las fuentes de datos. Existen dos tipos de fuentes disponibles en nuestro proyecto, internas o externas. A su vez, la información a obtener de estas fuentes puede tener diferente procedencia, como bases de datos, memoria SD del dispositivo móvil, etc. Por consiguiente, para cada tipo de procedencia de los datos, se debe implementar un método distinto de obtención de la información.

**Fuentes de información internas:** Nos referimos con fuentes de datos internas a aquellas fuentes que pueden ser accedidas desde el mismo dispositivo del móvil, sin tener que usar conexión ninguna. Son las fuentes que nutren a la aplicación Travel Blog para que sea capaz de funcionar en modo offline.

- Base de datos de Android:

SQLite es la tecnología de la base de datos que utiliza Android. Para adaptarla a nuestro proyecto y hacer mucho más sencilla la codificación de operaciones contra la base de datos, se ha optado por usar un mapeado de la base de datos objeto-relacional (ORM) mediante la librería ORMLite.

Esta librería nos proporciona una simple y ligera funcionalidad para la persistencia de objetos Java en bases de datos SQL, evitando la complejidad y la sobrecarga de otras librerías **ORM** como Hibernate.

ORMLite permite tratar los registros en la base de datos como si fueran objetos Java, lo que hace la vida de cualquier programador mucho más sencilla, y el diseño de consultas contra la base de datos al alcance de cualquiera. En la figura 2.5 se observa el esquema de funcionamiento de un mapeado objeto-relacional como ORMLite.



Figura 2.5: Arquitectura de ORMLite

- Tarjeta **SD** del móvil:

En muchos sitios está considerado una fuente de datos externa, pero al estar incluida dentro del dispositivo y no necesitar ninguna librería adicional especial para su acceso, en este contexto se va a considerar como una fuente de datos interna.

El caso de obtener información almacenada en la tarjeta **SD** del móvil se da, por ejemplo, cuando asociamos un archivo del repositorio local, una foto o un vídeo, a una entrada o recurso turístico. Para ello, Android proporciona al desarrollador una serie de consultas o llamadas al sistema operativo, conocidas como Intents y comentadas en la introducción de este documento en la sección 1.2.1, que dan como resultado la ejecución de una aplicación que permite la exploración de los ficheros de la tarjeta de memoria. Una vez elegido el archivo deseado, el mismo Intent que realiza la llamada, devuelve la información requerida correspondiente a dicho archivo.

- Software del móvil (Camera, Gallery):

Para poder obtener información a partir del uso de aplicaciones nativas de Android, se utilizan Intents predefinidos en el **SDK**, de una forma similar a como obtenemos información de la tarjeta de memoria **SD** del móvil. Éstos

Intents o mensajes llevan consigo información adicional que Android sabe interpretar ejecutando aplicaciones capaces de devolver el tipo de información requerida. Ejemplos usados en Travel Blog:

- `MediaStore.ACTION_IMAGE_CAPTURE`: se informa al sistema Android de que se quiere capturar una imagen con una aplicación instalada que permita hacerlo, como la cámara de fotos nativa del sistema operativo en el caso de Travel Blog.

- `Intent.ACTION_GET_CONTENT`: se informa al sistema Android que se quiere seleccionar un elemento de un tipo preestablecido anteriormente. Utilizado para seleccionar imágenes, vídeos, textos, etc. Esta llamada sirve del mismo modo para obtener información de archivos de tipo específico, como \*.gpx, \*.jpg, etc, de la tarjeta de memoria del dispositivo móvil.

- Otras aplicaciones (Google Calendar):

La única manera de conseguir una comunicación entre aplicaciones es mediante Content Providers. Android ofrece Content Providers para poder crear una comunicación con las aplicaciones nativas que vienen preinstaladas en cada dispositivo Android o con otras aplicaciones que permitan compartir información. En Travel Blog se aprovecha una de estas herramientas para poder acceder a la aplicación Google Calendar y tener la opción de poder crear un evento para cada una de las nuevas entradas de un cuaderno de viaje.

- Hardware del móvil (GPS): La API de Android ofrece al desarrollador librerías que se ponen en contacto con cualquier hardware que disponga el móvil, y del cual la aplicación tenga permiso de uso.

Estos permisos se incluyen dentro del archivo de configuración `AndroidManifest.xml`.

Ejemplos de estos permisos incluidos en el archivo de configuración son:

- `<uses-permission android:name="android.permission.INTERNET"/>`

- `<uses-feature android:name="android.hardware.location.gps"/>`

Cada uno de estos permisos tiene sus propias librerías en el SDK de Android para el tratamiento de la información obtenida.

El permiso al uso de hardware del dispositivo móvil es muy importante, ya que usado de un modo incorrecto podría causar diversos inconvenientes al usuario del móvil. Por ello, sólo al inicio de la primera ejecución de la aplicación, se debe preguntar al usuario sobre el uso permitido de los diferentes componentes hardware que van a ser utilizados por la aplicación.

### Fuentes de información externas:

Existen varios medios en los dispositivos Android para la comunicación con fuentes externas, como pueden ser el Bluetooth, red de datos, NFC<sup>5</sup>, etc. En Travel Blog

---

<sup>5</sup>Near Field Communication



únicamente vamos a utilizar el acceso a la red de datos como medio para la comunicación con fuentes externas. Para tener acceso a Internet desde la aplicación es necesario pedir permisos al usuario, antes de la instalación de la misma, para establecer conexión con el exterior.

- Mediante servicios web: En el apartado 1.2.4 “Contexto tecnológico” correspondiente a esta memoria, se ha realizado una breve explicación. Mediante servicios web, la aplicación móvil tiene acceso a la información de un almacén de datos externo, aunque no de forma directa. Para ello, la aplicación necesita conocer qué información nos puede dar dicho servicio web, y cómo la debemos pedir para conseguirla.

Como también se ha comentado, en la aplicación móvil únicamente establecemos comunicación con servicios web **REST**, servicios que atienden peticiones **HTTP**.

Para construir los mensajes que se van a intercambiar entre la aplicación y el servicio web, utilizamos el cada vez más popular formato **JSON**, un formato mucho más ligero que **XML** y que es tratado de una manera más eficaz. Además, contaremos con la ayuda de la librería GSON, una librería que crea y traduce automáticamente mensajes **JSON** a partir de objetos Java.

En nuestra aplicación se utiliza la comunicación con los servicios web cuando queremos obtener información de GeoSisTur, como por ejemplo confirmar las credenciales de un usuario, obtener cuadernos de viaje o listar **POI**<sup>6</sup>'s de una zona concreta.

También en la búsqueda de localizaciones geográficas sobre el mapa se obtiene información a través de un servicio web como es Google Geocoding<sup>7</sup>, que ofrece información geográfica de un lugar mediante la búsqueda de una entrada de texto. Es el método de búsqueda geográfica que utiliza Google Maps.

- Servidor de Tiles: La descarga y visualización de los mapas se hace a través de un servidor de *tiles*. Una *tile* es cada una de las partes gráficas que forman la visualización de un mapa. Cada *tile* es diferente y todas son del mismo tamaño. Como ya se ha comentado, para Travel Blog se ha elegido una librería que usa **OSM**<sup>8</sup>, una iniciativa cuyo objetivo es crear y proporcionar datos geográficos libres de todo el mundo. A través de esta librería, llamada Maps-Minus[11], la aplicación se conecta con un servidor para realizar la descarga de las *tiles* que componen la zona deseada de visualización de un mapa.

### 2.4.3. Dotación de interacción

Ya con el interfaz realizado, y la información requerida obtenida, comienza la dotación de interacción entre la ventana de navegación en cuestión y el usuario.

---

<sup>6</sup>Point Of Interest

<sup>7</sup><http://code.google.com/intl/es-ES/apis/maps/documentation/geocoding/>

<sup>8</sup>Open Street Maps

En Android hay más de un modo para capturar la interacción del usuario con la aplicación móvil. Cuando se habla de capturar la interacción, se refiere a la captura de eventos originados a partir de un objeto View con el que existe interacción por parte del usuario. La clase View proporciona todos los medios para conseguir ésto.

Para capturar esos eventos creados en la interacción, antes han tenido que tomar parte los llamados Listeners. Un Listener es una interfaz dedicada a detectar cualquier contacto sobre un *widget* o elemento de la interfaz en el que ha sido registrado dicho Listener. Para cada Listener registrado a un *widget*, existen métodos de activación.

Estos métodos son los que según el tipo de evento capturado por el Listener, realizará una acción u otra.

A cada método le corresponde un tipo diferente de evento, y cada evento está relacionado con un tipo diferente de gesto táctil del usuario. A continuación se presenta una breve descripción de los gestos táctiles más comunes con su método de acción correspondiente, y un ejemplo de uso en la aplicación Travel Blog:

- **Pulsación simple - método `onClick`**

Este evento(fig. 2.6(a)) es el más común de todos. Es activado al capturar un evento de pulsación simple con un solo dedo sobre uno de los elementos Android de la interfaz, como pueden ser botones, textos, imágenes, etc.

- **Pulsación larga - método `onLongClick`**

Este evento es similar al evento anterior, con la diferencia de que el ítem permanece pulsado durante un tiempo superior a un segundo. En Travel Blog es utilizado para mostrar un menú contextual. Este tipo de menú es el que aparece al realizar este tipo de evento táctil sobre uno de los items de un objeto View lista.

- **Al tacto - `onTouch`**

Este evento es capturado al notar cualquier tipo de tacto, incluyendo una pulsación, una liberación o cualquier gesto de movimiento sobre la zona en la que se encuentra el *widget* que registra este tipo de eventos.

En la aplicación este evento es utilizado en la galería multimedia. Nos permite la navegación de una imagen a través del tacto. Este evento se encarga de que, por ejemplo, sobre una imagen podamos realizar una ampliación del tamaño al tacto de dos dedos separándose hacia fuera (fig. 2.6 (d)), o una reducción acercando los dedos. También permite desplazar la imagen sobre la pantalla para, por ejemplo, visualizar zonas de la imagen no visibles debido a un alto nivel de zoom y al tamaño insuficiente de la pantalla.

- **Desplazamiento horizontal - `onFling`**

Este evento es capturado cuando se realiza una pulsación simple seguida de un movimiento horizontal con un solo dedo(fig. 2.6(b)).

En Android, el tamaño de pantalla de los dispositivos móviles muchas veces es insuficiente para la cantidad de información a mostrar. Por ello, este tipo de eventos es muy utilizado, ya que permite el desplazamiento de la pantalla y así poder visualizar información que se mantiene oculta en la interfaz de usuario. Otras estrategias para el aprovechamiento del espacio en Android se pueden ver en la sección 2.5.1.

En TravelBlog es utilizado en las galerías de imágenes y en la galería multimedia de la aplicación. El evento permite la navegación horizontal de las distintas imágenes de las que consta cada uno de dichos elementos.

#### ■ Desplazamiento vertical - onScroll

Este evento es capturado cuando se realiza una pulsación simple seguida de un movimiento vertical con un solo dedo(fig. 2.6(c)). En TravelBlog es utilizado en las listas de múltiples items y en distintas actividades como las que muestra la información detallada de un POI, donde no toda la información tiene espacio en la vertical para la visualización.

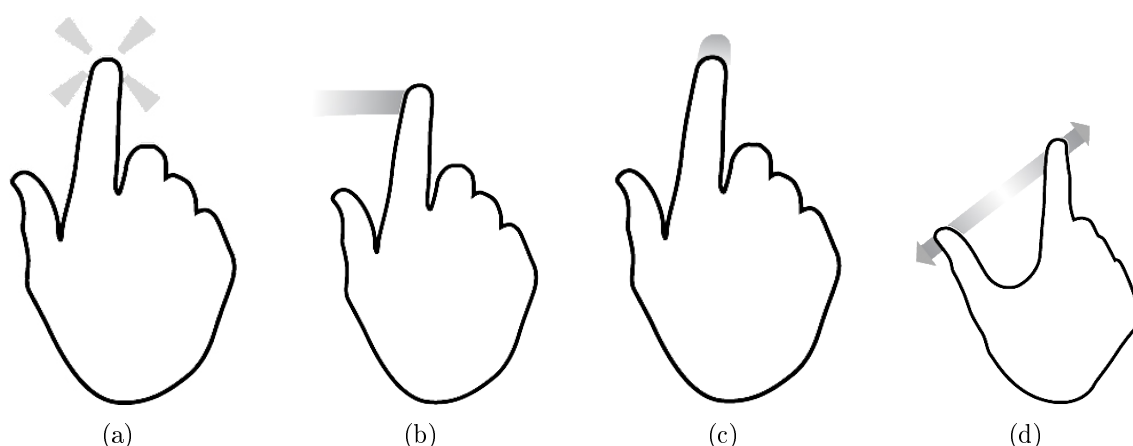


Figura 2.6: Acciones gestuales de interacción con Travel Blog: (a)-Pulsación simple; b()-Desplazamiento horizontal; c()-Desplazamiento vertical; d()-Aumento de nivel de zoom

Además de los métodos descritos anteriormente, existen muchos otros. Incluso se pueden crear Listeners a medida conteniendo más de uno de estos métodos básicos. Un ejemplo claro es visible en la interacción del usuario con los mapas. En Travel Blog, la Activity dedicada a la visualización del mapa, está registrada a un Listener a medida. En dicho Listener podemos observar algún que otro evento táctil diferente a los comunes:

#### ■ Pulsación simple - onSingleTapUp

Es similar, por no decir igual, que el evento básico cuyo método de acción es onClick. Su comportamiento y realización es el mismo. Con un solo dedo, este evento es disparado al pulsar y liberar la zona pulsada inmediatamente.

- **Pulsación doble - onDoubleTap**

Este evento es disparado cuando se realiza con un solo dedo y una doble pulsación simple en un breve periodo de tiempo sobre una misma zona, similar al doble click que se hace sobre el ratón de ordenador convencional. En Travel Blog se usa dentro de la visualización del mapa, donde nos proporciona un modo más de aumentar el nivel de zoom.

- **Pulsación larga - onLongPress**

Este evento es igual que el evento que acciona el método onLongClick, explicado con anterioridad. En el mapa, este evento nos permite centrar el mapa en la zona pulsada, y nos ofrece la capacidad de registrar como **POI** propio del usuario la zona pulsada.

#### 2.4.4. Pruebas de integración en la aplicación

Como última fase, una vez se tiene a cada elemento de la ventana dotado de interacción, procedemos a realizar las pruebas de su correcto funcionamiento.

Durante el desarrollo de la aplicación, básicamente se han realizado tres pruebas diferentes para cada una de las ventanas de navegación:

- **Prueba de cada *widget* independientemente.** Se prueba que la acción a desempeñar por dicho elemento se realiza correctamente sin ningún tipo de error.

- **Prueba de cada elemento cuidando el funcionamiento del resto.** Se prueba que la acción desempeñada individualmente por dicho elemento no afecta negativamente al comportamiento de ningún otro elemento presente en la interfaz.

- **Pruebas de integración de la ventana con el resto de ellas.** Estas pruebas se refieren a la comprobación del correcto paso de información de una ventana a otra a través de Intents.

Como un ejemplo concreto de la fase de pruebas realizadas, vamos a analizar un elemento de la aplicación, como es la galería de imágenes que desempeña la función de calendario y gestiona las entradas de un cuaderno de viaje. En este caso existen dos Listeners registrados en la galería. Uno de ellos controla la pulsación simple sobre cualquier ítem de la galería. El otro Listener controla que exista un ítem seleccionado en la zona central destacada de la galería.

La primera prueba consiste en que para cada una de las dos acciones, se miren los datos de las variables y se comparen con los datos que se esperan conseguir. Esto lo podemos realizar gracias a la herramienta de depuración disponible en el entorno de programación.

La segunda prueba consiste en comprobar que los demás elementos de la actividad funcionen de manera correcta. En nuestro caso se comprueba que para los dos tipos de eventos detectados por los listeners registrados en la galería, varios *widgets* cambien consecuentemente:

- La lista se carga con la lista de entradas pertenecientes al ítem de la galería seleccionado.
- La barra de texto que encabeza la galería contiene el año del día seleccionado.
- La barra de texto que encabeza la lista de entradas contiene el día y mes del ítem de la galería seleccionado.

Y como última prueba, se supervisa que el comportamiento dentro de la aplicación sea el correcto. Por ejemplo, se comprueba que la galería, al iniciarse la Activity que la compone, muestre el primer día del periodo que engloba la duración del cuaderno de viaje. De igual manera se comprueba que, volviendo a esta Activity desde una Activity posterior, el ítem que señala la galería es el mismo que se había seleccionado en última instancia.

Con estas tres pruebas se puede concluir que el funcionamiento del *widget* de la galería que realiza la función de calendario es el correcto y esperado.

En caso que durante las pruebas se encuentre algún fallo en la aplicación, el entorno utilizado en este **PFC** proporciona herramientas que ayudan a la localización de fallos en la ejecución. A través de su herramienta de depuración se puede ir ejecutando instrucción a instrucción el código fuente, comprobando el cambio en el estado de los registros, el valor de las variables, etc, en busca de fallos.

Aparte del depurador, Android ofrece una herramienta llamada LogCat. Esta herramienta proporciona al desarrollador un mecanismo para recolectar y ver los mensajes de salida del sistema de depuración. Siempre que algo extraño ocurra, se puede consultar LogCat en busca de un fallo inesperado.

## 2.5. Decisiones de diseño adoptadas

A lo largo de esta sección se van a exponer los puntos críticos que han surgido durante la realización de Travel Blog y las decisiones de diseño e implementación que se han tomado para cada uno de los casos.

### 2.5.1. Representación de la información

Uno de los principales problemas surgidos al inicio del proyecto, fue la falta de espacio para la representación de gran cantidad de información.

La aplicación a desarrollar está orientada para su ejecución en smartphones, lo que obliga a todo programador de aplicaciones para móvil a usar metodologías y patrones de diseño para realizar una adecuación apropiada del tamaño del dispositivo móvil a la cantidad de información a mostrar.

Además, por experiencias personales, se tiene muy en cuenta otros aspectos importantes como el tener toda la información necesaria accesible con el mínimo número de clicks, disfrutar de una navegación intuitiva y, pedir o mostrar la mínima información posible al usuario.

Para solucionar estos problemas, en Travel Blog se han utilizado algunas herramientas concretas proporcionadas por Android que se enumeran a continuación:

- Listas y galerías: En Android, uno de los *widgets* más potentes proporcionados son las listas. Son secuencias de items que tienen la posibilidad de mostrar diferentes *widgets* embebidos en cada uno de esos items, y que permiten un desplazamiento en vertical automático. Las galerías podrían definirse como listas horizontales, pero están más dedicadas a la visualización de fotografías y no son tan potentes como las listas. Pero ambas potencian el aprovechamiento del espacio gracias al desplazamiento automático del que gozan ambos *widgets*.
- Menús: Existen dos menús proporcionados por el **SDK** de Android. El menú de opciones, emergente en la zona inferior de la aplicación tras la pulsación del botón “Menú de Opciones”, y el menú contextual, disponible después de una pulsación prolongada de uno de los items de una lista. Ambos nos permiten ocultar opciones que, primando la muestra de información, son necesarias para el aprovechamiento funcional de la aplicación.
- Barra de acción: Se trata de una barra superior donde se muestra el título y un número limitado de opciones diferentes. En Travel Blog se ha utilizado la librería GreenDroid para la implementación de la barra, siendo de este modo, algo sencillo y de apariencia más que agradable para el programador. La barra de acción es aprovechada para incluir las opciones más importantes que deben estar presentes en la Activity, como por ejemplo en Travel Blog el acceso a la Activity del visor del mapa.
- Dialogs: Un Dialog es un objeto que es mostrado sobre una actividad, quedando de una manera flotante sobre ésta. Se puede decir que realiza el papel de ser una extensión de la pantalla. Los Dialogs permiten la muestra de una información adicional necesaria sobre la interfaz de usuario. En Travel Blog utilizamos Dialogs para mostrar alertas, presentar cuestionarios, expandir textos, seleccionar fechas, etc.

### 2.5.2. Calendario

Uno de los primeros retos a desempeñar durante el desarrollo de la aplicación, fue la creación del calendario gestor de las entradas de un viaje. El calendario era una parte clave de la aplicación y algo que podría diferenciar el modo de gestionar los viajes respecto a otras aplicaciones.

La primera opción que se tuvo fue usar el calendario que iba incluido como elemento Android, pero enseguida se rechazó por dos importantes razones:

- El calendario ocupaba toda la pantalla como si un casillero se tratara y no era escalable. Por lo tanto, no era lo que se buscaba, ya que en el diseño se había decidido mostrar el calendario además de otra información adicional.
- La mínima version Android necesaria para utilizar la vista de calendario incluida en el SDK era la 3.0 (Honeycomb), versión superior a la mínima pensada y además sólo soportada por tablets.



Figura 2.7: Calendario temporal inicial de Travel Blog

Como medida temporal, no queriendo que el desarrollo del calendario se convirtiera en un cuello de botella para el PFC, se decide utilizar una API ya creada que mostraba un calendario como una imagen, más bien estática, donde se representara la disposición de los días separados por semanas, para cada uno de los diferentes meses de un año, como si de una imagen clásica de calendario de pared se tratara.

Los inconvenientes de esta idea de calendario es que adolece de poca funcionalidad y muestra demasiada información para lo que realmente se ha establecido como necesario. Además, carece de usabilidad para la gestión de las entradas de un viaje. Su única función es señalar los días que comprenden un viaje. En la figura 2.7 se pueden observar el diseño inicial del calendario.

Más tarde, se toma la decisión de modificar el calendario de prueba. Se piensa en eliminar la información sobrante, ya que presentar todos los días de un mes resulta excesivo, y se piensa en dotarle de funcionalidad, mostrar la entradas correspondientes al día seleccionado.

Tras haber implementado galerías de imágenes en otras Activities, se piensa que por su diseño gráfico y usabilidad, se puede adecuar a la idea de calendario que se busca.

Para cada viaje, se crea una galería cuyas imágenes son los días pertenecientes a dicho viaje. La ayuda del scroll horizontal del que gozan las galerías Android hace que no sea un gran problema el gestionar un viaje de muchos días de duración. Además, estando la galería con un día seleccionado, se muestran en una lista inferior todas las entradas correspondientes a ese día.

Por lo tanto, se establece como solución final al calendario gestor de entradas ya que se cumplen los objetivos que se buscaban:

- Información necesaria.
- Tamaño apropiado a la interfaz.
- Funcionalidad deseada de la gestión de las entradas.
- Intuitivo, usable y agradable gráficamente.

La implementación de la solución final al calendario se puede observar en la figura 2.8.



Figura 2.8: Calendario creado para la gestión de las entradas de un cuaderno

### 2.5.3. Visor de mapas

Desde los inicios del proyecto, se establece que un visor de mapas es indispensable para Travel Blog. Es una herramienta de gran ayuda al turista que le permite ver en cada momento dónde se encuentra, localizar recursos y conocer el camino para llegar a un destino deseado.

Desde el punto de vista de un turista, un mapa es indispensable. Por ello se le da prioridad y fácil accesibilidad al mapa en cualquier ventana de la aplicación a partir de una opción sobre la barra de acción.



A continuación, se establecen unos requisitos al visor de mapas buscado para la aplicación:

- Mapa global: En la empresa donde se elabora el proyecto disponen unicamente de mapas propios de zonas concretas, como Zaragoza, Belchite, etc. Tratándose de Travel Blog, en principio, una aplicación móvil genérica, se busca que el visor de mapas muestre un mapa global, de todo el mundo.
- Mapa offline: Aparte de la visualización del mapa, la disponibilidad offline de éste se convierte en una funcionalidad muy deseada y que puede marcar diferencias con otras aplicaciones del mismo ámbito.
- Visor de mapas de código abierto: Bien es sabido que actualmente Google Maps goza de gran popularidad en todos los sentidos. Aunque también es cierto que el acceso a la API de Google Maps no sólo se ha convertido en una API de pago, si no que un desarrollador no tiene acceso al código fuente para crear mapas personalizados.

Una vez definidos los requisitos para el visor de mapas, se busca una API que se ajuste lo máximo posible a ellos.

Como elección, tras muchos análisis y pruebas con diferentes visores, se escoge una API llamada OSMDroid<sup>9</sup>. Se trata de un conjunto de herramientas para crear vistas e interactuar con mapas OSM sobre Android. Es una API de código abierto cuyo código fuente es de libre acceso. Ésto hace posible la personalización del mismo y el conocimiento de su funcionamiento, algo que valoro mucho personalmente, ya que se trata de un área desconocida.

Como fase de personalización del mapa, se han añadido los siguientes elementos acordes con Travel Blog, pudiéndose observar algunos de ellos en la ventana mostrada en la figura 2.9:

- Iconos personalizados para fotos, POI's, rutas, y otros elementos geolocalizados.
- Indicación del nivel de zoom en el que se encuentra el visor.
- Creación de un icono de posicionamiento sobre la pantalla para el registro de nuevos POI's.
- Búsqueda de lugares geográficos mediante entrada de texto.
- Muestra sobre el mapa de parte de la información de un elemento del usuario geolocalizado.
- Creación de un botón que indica la localización del usuario vía GPS o red de datos.
- Descarga del mapa visualizado en pantalla.

---

<sup>9</sup><http://code.google.com/p/osmdroid/>

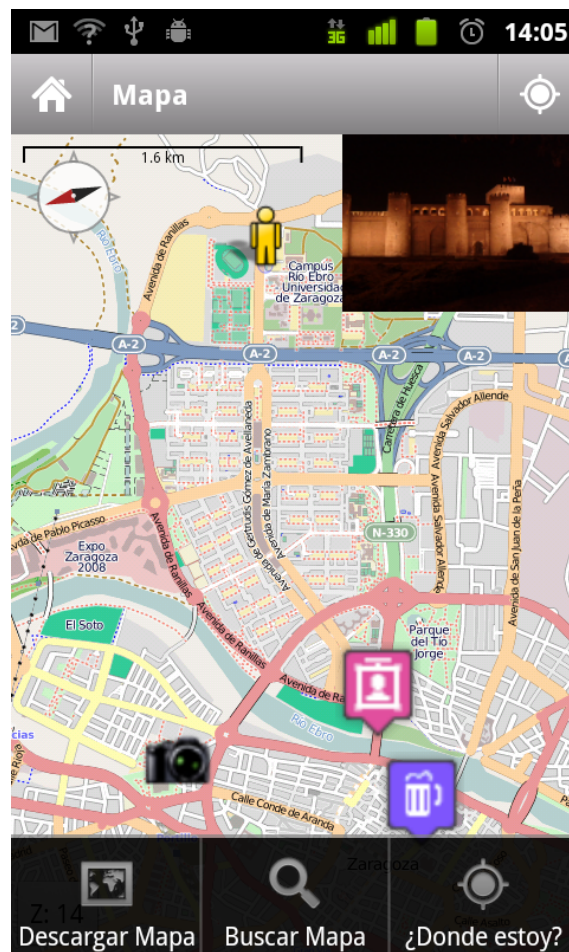


Figura 2.9: Visor de mapas modificado para Travel Blog

#### 2.5.4. Elementos multimedia

El tratamiento de los datos multimedia, sobre todo fotos y vídeos, es un tema delicado, ya que son grandes archivos que hay que transmitir entre la aplicación y GeoSisTur en ambos sentidos. Se necesita diseñar una solución que minimice la carga de datos pero sin que se vea afectado el contenido multimedia.

Otras aplicaciones móviles, como la popular WhatsApp<sup>10</sup> o la analizada previamente Tourist Eye, comprimen las fotos y los vídeos de tal manera que ocupen pocos kilobytes y puedan transferirse más rápidamente. La calidad, obviamente, se ve perjudicada, pero suficiente para la verse correctamente en una pantalla de teléfono móvil.

La decisión tomada por las dos aplicaciones anteriores ha sido también la elegida para Travel Blog. En el sentido de la aplicación hacia GeoSisTur, antes de enviarse mediante mensajes JSON, se comprimen los archivos multimedia a ser transferidos. En el sentido contrario, el servicio web de GeoSisTur es quien hace la misma tarea

<sup>10</sup><http://www.whatsapp.com/>

para transferirlos a la aplicación móvil.

### 2.5.5. Comunicación con GeoSisTur

Quizás, junto con el desarrollo del mapa de la aplicación, sea uno de los hitos más importantes que este PFC ha tenido que desempeñar. El diseño de la comunicación ha sido una tarea constante. Desde los primeros análisis, hasta la consecución de la aplicación, se ha ido entrando cada vez más en detalle. Pero no ha sido hasta haber finalizado con la parte offline de la aplicación, cuando la comunicación con GeoSisTur ha sido desarrollada.

Al igual que con los mapas, lo primero de todo es buscar los requisitos a cumplir con la comunicación mediante servicios web.

- Se desea un protocolo sencillo que no deba implementar mucho más de lo disponible en el SDK de Android.
- El formato de los mensajes debe ser ligero para ayudar a la fluidez de la comunicación.
- Mensajes de fácil tratamiento y que tengan algún tipo de soporte para trabajar con objetos Java.
- Comunicación síncrona.

Por el momento no se tiene pensado en crear una comunicación asíncrona con GeoSisTur, aunque se puede dejar para líneas futuras.

A continuación cito alguna de las principales conexiones con GeoSisTur:

- Login/Password.
- Nuevo registro de usuario.
- Actualización de cuadernos de GeoSisTur.
- Consulta y descarga de POI's de la comunidad GeoSisTur.
- Nuevo comentario.

La elección ha sido la de usar el protocolo REST, mediante paso de mensajes JSON con la ayuda de la librería GSON para la creación y la traducción de esos mensajes en objetos Java.

Con esta elección se cumplen todos los requisitos. En la figura 2.10 se puede observar un esquema de la arquitectura seguida para la confirmación de las credenciales en GeoSisTur.

A continuación se enumeran las principales razones por las que se ha elegido como servicio web que establezca la comunicación con GeoSisTur a un servicio REST.

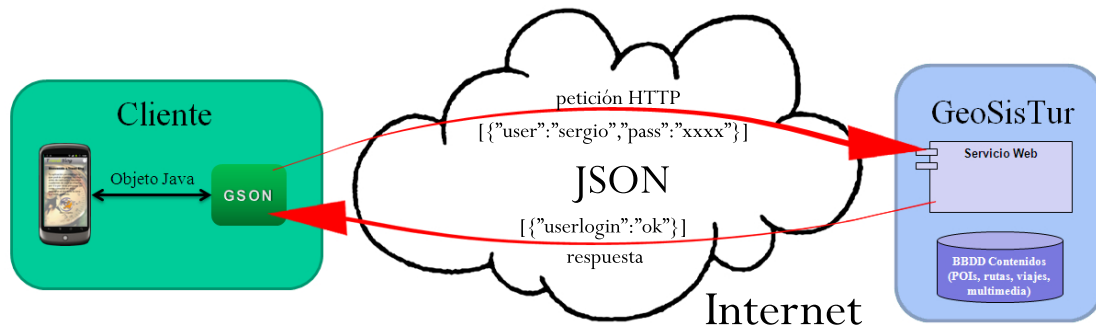


Figura 2.10: Arquitectura de la comunicación de Travel Blog con GeoSisTur

- Ligereza: no contiene demasiadas marcas XML.
- Resultados legibles por humanos.
- Fácil de construir tanto del lado del cliente como del lado del servicio, no necesita de herramientas especiales.
- Soporte incluido en el SDK de Android (HTTP).

Seguidamente se muestran las razones por las cuales se ha elegido JSON en detrimento de XML como el formato de los mensajes en el intercambio de datos con GeoSisTur.

- Utilizado y conocido dentro de la empresa.
- Formato ligero, compacto y eficaz para el intercambio de datos.
- Muy popular y extendido en el mundo de las aplicaciones para móvil.
- Fácilmente parseable con JavaScript mediante el comando eval().
- Soporte incluido en el SDK de Android.

Por último, se detallan las dos principales razones y ventajas que proporciona el uso de la librería GSON:

- Sencilla conversión de JSON a objeto Java: toJson() y fromJson().
- Soporte para objetos complejos.

# Capítulo 3

## Conclusiones

En este capítulo final de la Memoria se valorará el cumplimiento de los objetivos a la vista de los resultados obtenidos, se indicarán posibles líneas de trabajo como continuación del presente PFC, y se valorará la experiencia conseguida con su realización.

### 3.1. Resultados Obtenidos

El resultado más palpable obtenido con la realización del proyecto es la misma aplicación móvil genérica desarrollada durante el periodo que ha durado el mismo, sin olvidar la metodología que se ha seguido a la hora de la creación de aplicaciones móviles en Android, y el mundo de proyectos en esta novedosa área que se abren a partir de ahora.

Este PFC ha estado enmarcado en un contexto de empresa, incluido dentro del producto turístico ya mencionado anteriormente GeoSisTur. Considerando que GeoSisTur es una plataforma web genérica de creación de portales turísticos concretos, la aplicación que se ha desarrollado como resultado, puede adaptarse perfectamente a la personalización acorde a un portal turístico en concreto, debido a la modularidad e independencia de cada una de las partes del proyecto.

Por consiguiente se puede afirmar que se han cumplido los objetivos propuestos, ya que la aplicación es completamente funcional cumpliendo la totalidad de los requisitos y ya probada satisfactoriamente durante el disfrute de un viaje, su principal cometido.

### 3.2. Líneas Futuras

Durante la realización del proyecto han ido surgiendo múltiples vías de desarrollo que mejoren o añadan nuevas funcionalidades.

A continuación se muestra una lista de posibilidades a tener en cuenta en líneas futuras:

- Mejorar la interactividad con el mapa, ya que no se consideraba una parte esencial en la que profundizar durante el proyecto. Algunas ideas para esta mejora puede hacerse añadiendo nuevas fuentes de mapas independientemente del visor, incluyendo la creación y gestión de rutas, o añadiendo nuevas herramientas como indicar el camino a un destino, etc.
- Al aspecto gráfico no se le ha dedicado demasiado tiempo en la realización del proyecto. Aunque se ha intentado cuidar desde el principio, un diseño más cuidado y profesional dotaría a la aplicación de gran atractivo al usuario.
- Mejorar la usabilidad en cuestiones de carga de contenido. A la hora de cargar elementos multimedia como los son las fotos y los videos, esta tarea se realiza ítem a ítem. Una selección múltiple multimedia permitiría crear álbumes y daría más versatilidad a la aplicación.
- Otorgar al usuario mayor control de la aplicación añadiendo campos en la configuración de la misma, como por ejemplo elección de idioma, conexión y desconexión del **GPS**, o importación de viajes en algún formato popular.
- En el caso de que el producto tenga una gran adaptación, que la aplicación llegue al máximo número de usuarios es muy importante. Por ello, la adaptación a otras plataformas móviles como las tabletas, o el desarrollo de la misma para otros sistemas operativos móviles como puedan ser iPhone o Windows Mobile, incluso la internacionalización de la misma, ayudaría a conseguirlo.

### 3.3. Valoración Personal

La realización de este Proyecto Fin de Carrera ha tenido para mí repercusiones muy positivas, tanto a nivel profesional como personal.

En lo relativo al aspecto profesional, me ha brindado la oportunidad de poner en práctica conocimientos que se adquieren en las asignaturas cursadas a lo largo de la carrera, pero que no se tiene la ocasión de utilizar en un proyecto de larga duración, en el que se lleve una continuidad de principio a fin.

Me ha permitido tener una visión global de lo que es un proyecto software, y utilizando para ello multitud de herramientas que he ido adquiriendo en experiencia durante mi paso por la carrera, y que pensaba que nunca las iba a necesitar.

Además, la ventaja de poder desarrollar el proyecto en un ámbito empresarial aporta una experiencia añadida de lo que es el trabajo en un entorno laboral real, sintiendo el apoyo y consultando a tus compañeros, pues es imprescindible trabajar en grupo adecuadamente para llevar a buen término el proyecto.

Por otra parte, el hecho de desarrollar un producto real utilizado por un gran número de usuarios y como parte de otro producto ya afianzado en la empresa como es GeoSistur, aporta una gran satisfacción personal que permite enfrentarse con garantías a los problemas que puedan surgir en futuros proyectos.

En última instancia, el realizar un proyecto que guste personalmente, tenga tantas oportunidades como tiene, te permita estar en la cresta de la ola de la tecnología y crear para tí y para los demás, es algo que tuve muy en cuenta a la hora de elegir el proyecto. Ahora, después de la realización del proyecto, la ilusión por seguir el camino empezado permanece viva y con más ganas por aprender y seguir aprendiendo.

## Parte II

### Anexos



# Anexo A

## Gestión del proyecto

Para todo proyecto software de un tamaño considerable, es necesario utilizar técnicas de gestión que nos permitan planificar y llevar a cabo correctamente cada una de las etapas del proyecto en el tiempo previsto, sobre todo en proyectos comerciales con un plazo de entrega estipulado con el cliente. En los siguientes apartados se detallará la metodología de trabajo seguida en el presente proyecto, la planificación realizada y las herramientas utilizadas.

### A.1. Metodología de trabajo

En la empresa GeoSpatiumLab y en el grupo IAAA se desarrollan gran cantidad de proyectos en equipo, en la mayoría de las ocasiones de un tamaño bastante superior al de un **PFC**, situación en la que se hace imprescindible el uso de alguna metodología de trabajo que permita estructurar y organizar el trabajo a realizar. Al comienzo de este **PFC**, dentro del equipo de trabajo, ya existían algunas metodologías o guías para abordar algunos de los posibles problemas de gestión, propuestas por miembros del grupo con amplia experiencia en desarrollo de proyectos software, que han contribuido a mejorar la gestión del actual proyecto.

GeoSpatiumLab posee la certificación ISO<sup>1</sup> 9001 y 14001 y el nivel 2 CMMI<sup>2</sup>, y durante el mes de Octubre de 2011, durante la realización de este **PFC** se obtuvo el nivel 2 de la norma ISO/IEC 15504-5 (SPICE). Para el desarrollo de los proyectos se utiliza una metodología propia basada en metodologías orientadas a objeto. En el caso particular de este proyecto se siguió un proceso iterativo incremental dividido en fases:

- Análisis inicial.
- Diseño de la aplicación.
  - Creación de las ventanas de navegación

---

<sup>1</sup>International Organization for Standardization

<sup>2</sup>Capability Maturity Model Integration

- Diseño del modelo de datos.
- Elaboración de cada una de las pantallas. Cada iteración de elaboración de una pantalla comprende a su vez:
  - Análisis de requisitos de la pantalla
  - Diseño gráfico de cada una de las pantallas.
  - Implementación de la interacción.
  - Pruebas parciales sobre las pantallas.
  - Pruebas de integración de pantallas en la aplicación.
- Instalación y pruebas en host de producción.

El análisis, diseño, implementación y pruebas parciales de los módulos se realizaron en la máquina local, utilizando un dispositivo móvil específico para la depuración y realización de las tareas de integración.

Durante todas las etapas del desarrollo del proyecto se ha realizado un control de los esfuerzos dedicados, un control de versiones de los elementos generados y copias de seguridad de los mismos. Un aspecto a destacar ha sido la realización de frecuentes reuniones con otros miembros del equipo, con objeto de concretar los requerimientos de los diferentes componentes a implementar, interfaces de comunicación y tecnologías a utilizar de cara a facilitar todo lo posible la integración.

## A.2. Planificación

Para el presente proyecto se ha tenido que cumplir una planificación más bien libre en cuanto a fechas de entrega, al ser Android una nueva tecnología a tratar en la empresa, aunque se ha cuidado su desarrollo ya que ha sido incluido dentro de un producto comercial real como es GeoSisTur.

A continuación se enumeran cada a una las tareas en mayor detalle que se han realizado para conseguir los objetivos perseguidos en este **PFC**.

- Búsqueda y lectura de formación Android.
- Análisis de GeoSisTur y otras aplicaciones.
- Definición de los requisitos de la aplicación.
- Diseño de la aplicación.
  - Arquitectura del sistema.
  - Prototipado de las ventanas.
  - Modelo de datos.
- Diseño de las ventanas de la aplicación.
- Prueba e implementación de elementos Android.
- Desarrollo de la aplicación
  - Implementación de listas compuestas.
  - Creación de gestor de base de datos SQLite.

- Implantación de ORMLite.
  - Utilización de barras de acción.
  - Analisis y desarrollo del calendario.
  - Gestión multimedia.
  - Analisis de opciones en visualización de mapas geográficos.
  - Implementación y desarrollo del visor de mapas.
  - Mejora del calendario inicial.
  - Comunicación con servicios web.
  - Realizar pruebas de integración.
- Documentación
  - Realización de la memoria.

Teniendo en cuenta que son aproximadamente seis meses de trabajo, y que se comenzó a principios de julio, la fecha de entrega quedó establecida en febrero. Además, en la empresa cuentan con un sistema de gestión de proyectos en el que se incluye un estricto control de esfuerzos que me permite conocer el tiempo real en horas dedicadas a este PFC. El total de horas dedicadas durante estos seis meses ha sido de alrededor de mil horas. En la figura A.1 se muestra un diagrama de Gantt con la planificación en periodos dedicados para las distintas tareas.

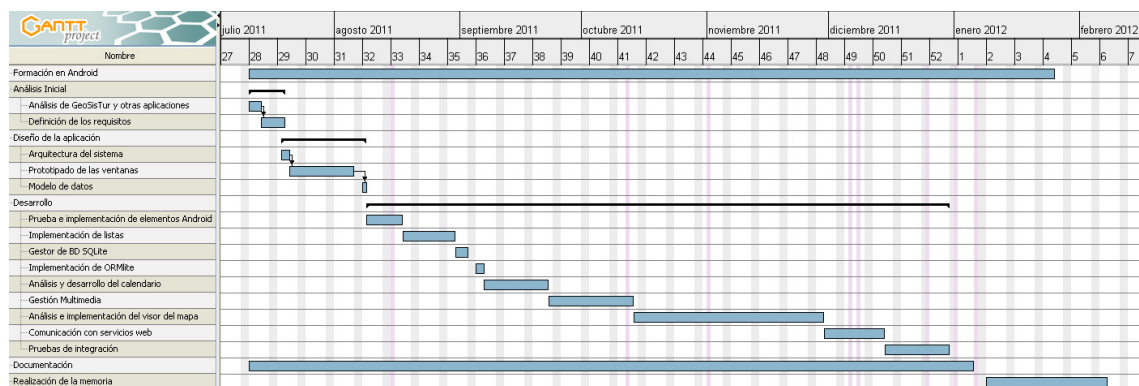


Figura A.1: Cronograma final del PFC.

### A.3. Herramientas utilizadas

Este proyecto ha sido programado principalmente en lenguaje Java, lenguaje en el que está basado Android, utilizando Eclipse<sup>3</sup> como principal herramienta de trabajo.

Eclipse es una plataforma de desarrollo Open Source y multiplataforma basada en Java. La característica más destacable de Eclipse es la extensibilidad, ya que es una gran estructura formada por un núcleo y múltiples plugins que van conformando

<sup>3</sup><http://www.eclipse.org/>

la funcionalidad final. La forma en que los plugins interactúan entre sí es mediante interfaces o puntos de extensión; de esta forma las nuevas aportaciones se integran sin dificultad. Otra gran ventaja para la realización del proyecto es la integración de Eclipse con el sistema de versionado Subversion, que además sirve para la realización de copias de seguridad de los documentos y para la consulta del proyecto por parte de la empresa.

Para la elaboración de este documento se ha utilizado L<sup>A</sup>T<sub>E</sub>X, un procesador de textos basado en un lenguaje de marcado. Permite escribir documentos centrándose en el contenido sin tener que preocuparse en el formato del mismo, como suele pasar en herramientas WYSIWYG —es decir, “Lo que ves es lo que obtienes”—, donde se requiere de una alta habilidad para el formateo de documentos, labor que se complica exponencialmente conforme va aumentando el tamaño del documento. Gracias a L<sup>A</sup>T<sub>E</sub>X este documento ha podido ser confeccionado de forma eficiente y con resultados visuales finales de alta calidad.

Además, para la edición, la confección de la memoria de este PFC se ha realizado con un plugin para Eclipse que proporciona soporte para L<sup>A</sup>T<sub>E</sub>X llamado T<sub>E</sub>Xlipse.

Otras herramientas utilizadas durante el desarrollo del proyecto han sido:

- **Evolus Pencil**: Diseño de ventanas Android.
- **Microsoft Office Visio**: Creación de diagramas de clases.
- **Microsoft Excel**: Control de esfuerzos diario.
- **Microsoft Word y PowerPoint**: Elaboración de la documentación generada en el Grupo y creación de diagramas de arquitectura.
- **Photoshop**: Herramienta gráfica para creación y edición de las imágenes utilizadas en la documentación, y para la creación de iconos y demás imágenes necesarias en la aplicación.
- **GanttProject**: Diagramas de Gantt usados para la planificación del proyecto.

# Anexo B

## Análisis de Requisitos

En la sección 1.3 del capítulo 1 de introducción, y con algo más de detalle en la sección 2.2.2, se han expuesto los requisitos que se determinaron para este proyecto en el momento de su concepción. Este anexo simplemente pretende dar una descripción más detallada de las características que debe cumplir el sistema software a construir.

### B.1. Requisitos funcionales

Para la plataforma tecnológica objetivo de este proyecto se tiene los siguientes requisitos funcionales:

**RF-1** Gestión de encuestas.

**RF-2** Funcionamiento offline.

**RF-3** Muestra de los cuadernos de viaje por orden de fecha de acción. Los cuadernos que se muestran en la aplicación pueden ser propios del usuario (editables) o ajenos al usuario(no editables), descargados a través de GeoSisTur.

**RF-4** Gestión de cuadernos de viaje (Alta/Baja/Modificación).

**RF-5** Un cuaderno está formado por título, fecha de inicio, fecha final, descripción del cuaderno, imagen de portada y entradas asociadas.

**RF-6** Mostrar información asociada a un cuaderno de viaje.

- Título.
- Imágen de portada.
- Fechas de inicio y final.
- Descripción.

- Entradas asociadas por día de cuaderno.

**RF-7** Gestión de entradas (Alta/Baja/Modificación).

**RF-8** Una entrada está formada por título, fecha de inicio, hora de inicio, descripción de la entrada, imágenes, vídeos, rutas y recursos asociados.

**RF-9** Mostrar información asociada a una entrada.

- Título
- Fecha y hora de inicio.
- Imágenes y vídeos asociados.
- Recursos asociados

**RF-10** Creación de eventos en Google Calendar a partir de la creación de una nueva entrada.

**RF-11** Gestión de recursos turísticos.

**RF-12** Un recurso turístico está formado por nombre, dirección postal, dirección web, teléfono de contacto, tipo de recurso, georeferencia (latitud,longitud) y imágenes, vídeos, comentarios y puntuación asociada.

**RF-13** Mostrar información asociada a un recurso.

- Teléfonos de contacto.
- Georreferenciación en el mapa.
- Indicaciones para llegar desde la posición actual.
- Página web.
- Valoración de los comentarios asociados al recurso.
- Comentarios asociados al recurso.

**RF-14** Un recurso puede ser creado por el usuario (editable) u obtenido a través de GeoSisTur (no editable).

**RF-15** Comentar y valorar recursos turísticos.

**RF-16** Creación y asociación de imágenes, vídeos o recursos turísticos nuevos o ya existentes a un cuaderno, entrada o recurso.

**RF-17** Asociación de rutas ya creadas a una entrada.

**RF-18** Gestión de recursos multimedia.

**RF-19** Muestra de las propiedades de un elemento multimedia.

**RF-20** Reproducción de multimedia y compartición por medio de e-mail, Bluetooth y otros medios disponibles en el dispositivo móvil.

**RF-21** Mostrar información geográfica sobre un mapa accesible desde un cuaderno, una entrada o un recurso de la aplicación.

- Localización de recursos multimedia y turísticos georreferenciados.
- Visualización de información de recursos, una pequeña imagen en el caso de recursos multimedia, y el título en el caso de recursos turísticos.
- Localización del usuario.
- Visualización del nivel de aumento de un mapa.
- Descarga de tiles online para la visualización de un mapa de una zona concreta offline.
- Muestra de una brújula.
- Búsqueda de lugares geográficos.
- Creación de lugares señalados sobre el mapa.
- Muestra de los **POIs** creados por el usuario.

**RF-22** Conexión e intercambio de datos (actualización, descarga y consulta) con GeoSisTur.

- Búsqueda y descarga de cuadernos propios y ajenos.
- Búsqueda y descarga de recursos turísticos.
- Actualización de cuadernos de viaje del usuario.
- Creación de **POIs** a partir de **POIs** del usuario.
- Actualización de comentarios sobre un recurso.
- Registro de nuevo usuario GeoSisTur.
- Gestión del perfil de usuario.

**RF-23** Mostrar Ayuda al inicio de la aplicación.

## B.2. Requisitos no funcionales

La plataforma tecnológica objeto del presente proyecto tiene los siguientes requisitos no funcionales:

**RNF-1** Creación de aplicación móvil para la plataforma Android de Google.

**RNF-2** Uso de tecnologías *Open Source* (Código Abierto) en el visor de mapas, concretamente OpenStreetMaps (**OSM**).

**RNF-3** Gestión y persistencia de objetos Java en base de datos SQLite a través de ORMLite.

**RNF-4** Conexión con servicios web externos mediante protocolo **REST** y formato de mensajes **JSON**.

# Anexo C

## Diseño y Modelo de datos

En este anexo se recoge todo lo que engloba la fase de diseño de la aplicación tras la obtención de los requisitos funcionales y se explicará con más en detalle la fase del diseño de las ventanas y la navegación a través de las mismas, así como el modelo de datos utilizado para la estructura de información necesaria en Travel Blog.

### C.1. Diseño de la aplicación

Durante la fase de diseño de la aplicación se han realizado dos tipos diferentes de tareas, una a continuación de la otra. La primera tarea ha sido la obtención de las ventanas de navegación de forma gráfica a partir de la herramienta de creación de prototipados de interfaces gráficas. La segunda tarea ha consistido en, a partir de las ventanas diseñadas, crear la navegación entre las mismas.

#### C.1.1. Prototipado de las ventanas de navegación

Evolus Pencil es la herramienta usada para esta tarea. Se trata de una herramienta de software libre para la elaboración de diagramas y de prototipado de interfaces de usuario. Durante el inicio de la fase de diseño se buscó una herramienta que permitiera crear pantallas Android, y Pencil cumplía todos los requisitos deseados.

Pencil admite juegos de formas prediseñadas para la elaboración de diagramas de distintos tipos. El hecho es que uno de estos juegos cuenta con todas las formas básicas de diseño en Android y, además, al tratarse de un software WYSIWYG, el trabajo con Pencil se hace muy versátil y ameno.

Pencil también permite varias formas de exportación del prototipado diseñado. Los resultados de dicha importación pueden ser en forma de imágenes independientes, ideal para incluir en documentación como en el caso de la memoria de este [PFC](#),



como también en forma de presentación HTML, realmente útil para la presentación al cliente de la navegación de las ventanas con un resultado muy profesional.

Tal ha sido el descubrimiento de Pencil a la hora de crear el prototipado de las ventanas de una aplicación Android, que se ha adoptado dentro de la empresa para el uso en nuevos proyectos del mismo tipo.

En la figura C.1 pueden verse algunos de los prototipos de ventana creados para el desarrollo de la aplicación Travel Blog.



Figura C.1: Prototipos realizados con Pencil en la fase de diseño

### C.1.2. Navegación de la aplicación

Tras la creación de los prototipos de pantalla de la aplicación, se realizó la conexión entre éstos. Aunque esta tarea se podía intuir de una manera aproximada, no fue tras el diseño de las pantallas cuando se estableció un plan de navegación fija para Travel Blog.

En la figura C.2 se puede ver la navegación básica entre los diferentes prototipos de pantalla diseñados.

## C.2. Modelo de datos

Una vez diseñadas las pantallas de navegación, necesitamos estructurar la información que éstas contienen mediante un modelo de datos. Dicho modelo es el que se puede ver en la figura C.3.

A partir del modelo de datos diseñado a más alto nivel, se ha ido buscando la cantidad mínima necesaria de información para cada una de las entidades del modelo, con el fin de dar una imagen de ligereza y usabilidad a Travel Blog. Podemos ver un esquema de los atributos que forman cada una de las clases en la figura C.4.

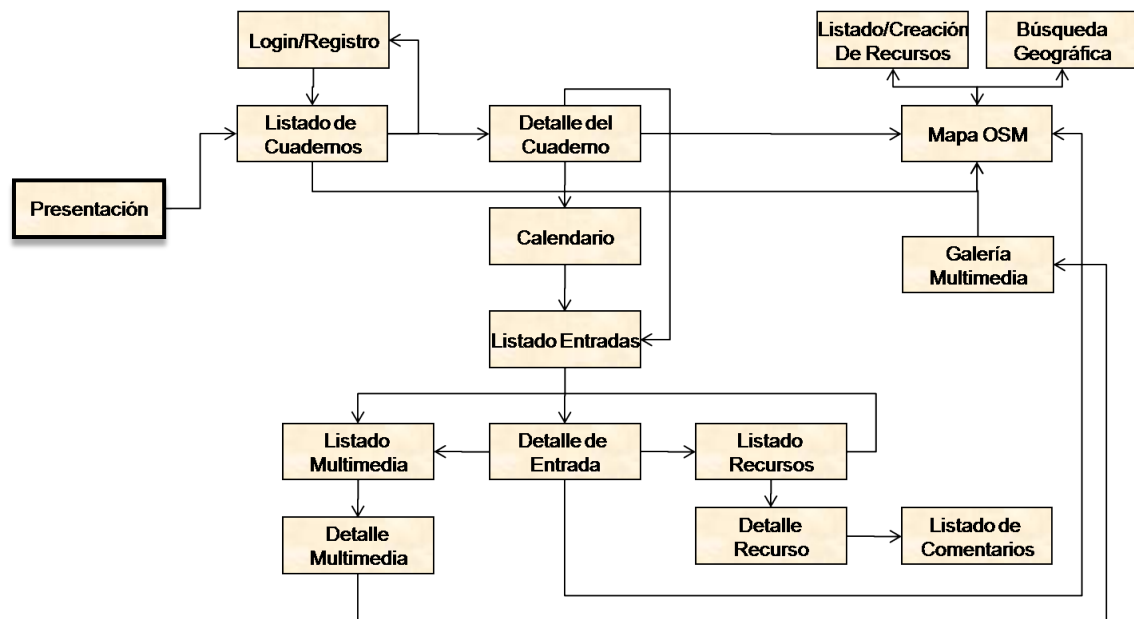


Figura C.2: Navegación de ventanas entre las principales pantallas diseñadas

La gran parte de los atributos son de sencilla explicación, pero existen algunos que merecen algo más de detalle:

- *editable*: Atributo de las entidades TravelBlog y Resource. Se refiere al permiso que indica si es posible la modificación o agregación de la información en cada una de las dos entidades.
- *flag*: Atributo de la entidad TravelBlog que nos permite conocer si un cuaderno de viaje es nuevo o, por el contrario, ya existe y se va a proceder a su modificación. Su cometido es el de aprovechar la interfaz gráfica y poder unificar la pantalla de creación con la pantalla de modificación de cuadernos.
- *path*: Presente en muchas de las entidades del modelo, representa la localización de una imagen o video dentro de la memoria del dispositivo móvil. Nos permite no tener que cargar con todo el peso del archivo cada vez que sea utilizado, si es que éste se encuentra en memoria.
- *imgrec*: Atributo de la entidad Resource, la cual denota el identificador de imagen representativa de cada tipo de recurso. En Android, las imagenes predefinidas añadidas al proyecto pueden ser accedidas gracias a que disponen de un identificador asignado en el proceso de compilación de la aplicación. Dicho identificador es el indicado por este atributo y, a su vez, el indicado por el atributo *rec* de la entidad POIType.



## Anexo D

### Arquitectura del sistema

En este anexo se explica más en profundidad la arquitectura pensada para la aplicación Travel Blog, donde se detalla la interacción con la estructura que ofrece el sistema operativo Android y las comunicaciones con las fuentes de información externas.

En la figura [D.1](#) se puede observar el diagrama con la mayoría de los componentes del sistema operativo Android.



Figura D.1: Arquitectura del sistema operativo Android

Básicamente, la arquitectura del sistema operativo Android se divide en cinco partes diferentes:

- *Applications* En esta parte se sitúan todas las aplicaciones instaladas por el usuario y las nativas del sistema. Todas estas aplicaciones están codificadas en lenguaje de programación Java. Es en esta zona está situada la aplicación desarrollada en este [PFC Travel Blog](#).
- *Application Framework* Es la zona del sistema operativo donde se encuentran las herramientas que Android proporciona al desarrollador de aplicaciones. Se podría decir que esta zona de la arquitectura se encarga de la conexión entre la aplicación y el exterior a través de diferentes interacciones soportadas por Android como sensores en pantalla, antena GPS, etc. Alguna de las herramientas incluidas en el *Application Framework* se enumeran a continuación:
  - Gestor de Activities.
  - Sistema de Views.
  - Gestor de recursos (iconos, audio, estilos, etc).
  - Gestor de localización.
  - Gestor de telefonía.
- *Libraries* Android incluye un conjunto de librerías localizadas en esta parte de la arquitectura usadas por diferentes componentes del sistema. Estas capacidades se le proporcionan al desarrollador a través del Application Framework. A parte de éstas librerías, en el desarrollo de Travel Blog hemos necesitado otras ajenas al sistema Android como por ejemplo:
  - ORMLite: usada para el mapeo objeto-relacional de la base de datos SQLite.
  - MetadataExtractor: utilizada para obtener información detallada de archivos multimedia como la orientación, la posición geolocalizada, etc.
  - GeoJson: librería utilizada junto a JSON, proporcionada por Android, que nos permite la conversión y traducción de objetos Java en mensajes JSON para el intercambio de datos con servicios web.
  - OSMDroid: permite la visualización básica de mapas [OSM](#) en dispositivos Android. Esta librería ha sido editada para conocer el funcionamiento interno y la modificación de la misma de acuerdo a los requisitos elegidos para el visor de mapas de la aplicación.
  - GreenDroid: librería que ofrece al usuario un ahorro de tiempo y facilidad en la creación de la interfaz gráfica de Android. En Travel Blog se ha utilizado únicamente para la creación de la barra de acción presente en muchas de las pantallas de la aplicación.

- *Linux Kernel* Android se basa en la versión 2.6 de Linux para los servicios básicos del sistema como la seguridad, la gestión de memoria, gestión de procesos, pila de red, y el modelo de controlador. El núcleo también actúa como una capa de abstracción entre el hardware y el resto de la pila de software.
- *Android Runtime* En esta zona se encuentran las librerías necesarias para dar funcionalidad a todas aquellas librerías proporcionadas por el lenguaje de programación Java. Cada aplicación Android ejecuta su propio proceso con su propia instancia de máquina virtual Dalvik, la cual ha sido construida para que el dispositivo pueda ejecutar multiples máquinas virtuales de un modo eficiente, y que podemos encontrar en esta parte de la arquitectura. La máquina virtual Dalvik es capaz de ejecutar las clases creadas por un compilador Java.

A toda la arquitectura del sistema Android, se deben añadir las diferentes fuentes de información con las que la aplicación interactúa. Por ello, en el esquema global de la arquitectura de la aplicación se añaden tanto las bases de datos utilizadas, como los servicios web mediante los cuales se comunica con repositorios de información ajenos al dispositivo, así como las conexiones con cada una de las partes de la arquitectura.

Como resultado final de la fase del diseño de la arquitectura de Travel Blog, en la figura D.2 se muestra un diagrama que plasma la estructura de la aplicación.

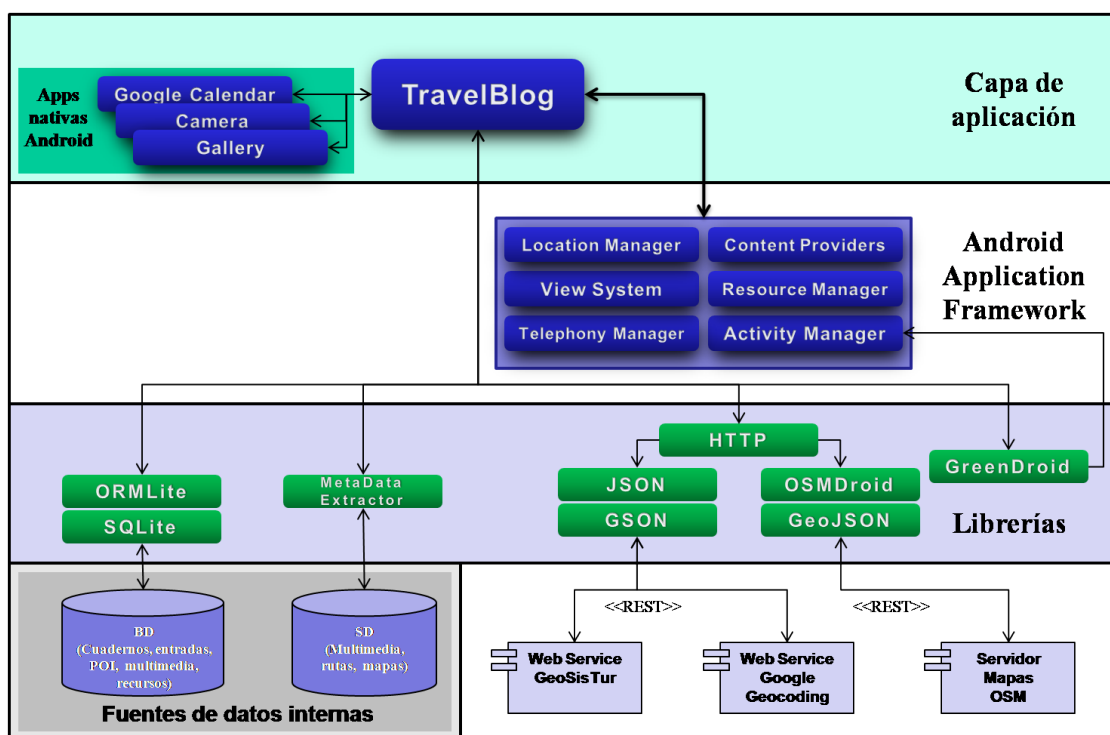


Figura D.2: Arquitectura de la aplicación Travel Blog

## Anexo E

# Manual de instrucciones Travel Blog

En este apéndice se recoge el manual de usuario de la aplicación móvil Travel Blog para Android desarrollada en este [PFC](#) como herramienta de ayuda en la introducción y gestión de los cuadernos de viaje, así como referencia para las tareas de mantenimiento más importantes.

**Manual de Usuario de la aplicación Android para móvil**

# **Travel Blog**



## Tabla de contenido

<b>¿Qué es Travel Blog? .....</b>	<b>3</b>
Instalación e inicio.....	4
 <b>Cuadernos de viaje.....</b>	<b>5</b>
Creación de un cuaderno .....	5
Descarga .....	6
Modificación .....	6
Detalles de cuaderno .....	7
Actualización.....	8
 <b>Entradas de cuaderno de viaje .....</b>	<b>8</b>
Creación de una entrada .....	9
Modificación .....	10
Detalles de Entrada .....	11
 <b>Detalle de recurso turístico .....</b>	<b>12</b>
 <b>Galería Multimedia .....</b>	<b>13</b>
 <b>Mapa .....</b>	<b>14</b>
Descargar Mapa .....	15
Búsqueda en el Mapa.....	15
Creación de POI.....	16
 <b>Conexión con GeoSisTur .....</b>	<b>18</b>
 <b>Pantallas de Travel Blog .....</b>	<b>19</b>

## ¿Qué es Travel Blog?

Travel Blog es una aplicación creada para todo móvil con sistema operativo Android.

Con Travel Blog serás capaz de gestionar tus viajes con todos los recursos necesarios de tres maneras diferentes:

- Planificando tu viaje antes de realizarlo.
- Plasmando tus experiencias a medida que transcurre el viaje.
- Completando tu cuaderno de viaje una vez concluido.

Travel Blog es la herramienta ideal para poder planificar cualquier escapada, viaje o trayecto. Tú podrás predefinir todos los eventos, lugares, visitas que no quieres perderte para tu escapada. Además, puedes utilizar las experiencias de otros usuarios a través de los cuadernos de viaje creados por ellos mismo.

A medida que vayas viajando e inmortalizando la experiencia a través de fotos, videos o rutas, seguro que encuentras un momento para ir completando tu cuaderno con el único fin de que no sea pasto del olvido y puedas compartirlo con otros viajeros.

Esto es posible gracias a que Travel Blog cuenta con una plataforma web llamada GeoSisTur, con la cual todas las experiencias que plasmes en el móvil, podrás verlas en Internet, y viceversa.

En Travel Blog pensamos en ti, y por ello podrás disfrutar en cada momento de un mapa del mundo entero con el que podrás cubrir las necesidades más básicas necesarias para un viajero como tú, por ejemplo:

- ¿Dónde me encuentro?
- ¿Dónde se encuentra esta ciudad?
- La torre Eiffel está aquí, ¡¡me lo apunto!!

Y para terminar, y lo más importante. ¿Alguna vez has viajado al extranjero o a alguna zona sin cobertura y no has podido conectarte con Internet, convirtiendo en inútiles algunas de tus aplicaciones? Con Travel Blog ésto no te pasará.

Utiliza Travel Blog donde quieras y cuando quieras, sin estar atado a conexiones de red. Travel Blog está preparado para ofrecerte toda su funcionalidad sin Internet.

Lo único que tendrás que hacer, es preparar tu viaje con antelación descargando mapas, rutas, lugares a visitar, etc, y olvidarte de buscar una conexión donde quiera que te encuentres.

¿Te animas a probar Travel Blog en tu próxima escapada?

## Instalación e inicio

Lo primero que deberás hacer para disfrutar de Travel Blog, es acudir al Market de Android o visitando GeoSisTur para poder descargar tu aplicación. Una vez descargada en tu dispositivo Android, sólo deberás instalarla y aceptar los permisos necesarios como cualquier otra aplicación, y listo, puedes comenzar.

Con lo primero que te encontrarás será con la pantalla inicial de presentación. Desde ella podrás acceder a tus cuadernos de viaje, conocer el funcionamiento básico de Travel Blog y tener una vía de contacto directa con el equipo técnico en caso de dudas, sugerencias o fallos en la aplicación.

Para acceder a tus cuadernos de viaje, tan sólo deberás pulsar el botón de inicio. Tras ello, durante un breve periodo de tiempo se iniciarán las tareas de creación de directorios e imágenes necesarias para el correcto funcionamiento de la aplicación. Y ya está, empezar con Travel Blog es muy fácil.



## Cuadernos de viaje

Ya hemos comentado que un cuaderno de viaje va a ser la entidad principal de la aplicación. En ella almacenaremos toda la información posible sobre cada una de nuestras escapadas.

Inicialmente, nuestra lista de cuadernos de viaje se encontrará vacía al acabar de iniciar la aplicación. Para poder crear o disponer de nuestros cuadernos de viaje tenemos dos opciones que veremos en detalle más adelante, como son la creación desde cero de un cuaderno de viaje y la descarga de cuadernos alojados en GeoSisTur.

### Creación de un cuaderno

Para la creación de un cuaderno de viaje existen dos modos de realizarlo: a través de la opción “MAS” en la barra de acción de la aplicación, o a través de “Nuevo Cuaderno” en el menú de opciones.

Cada una de estas dos maneras de crear un cuaderno nos llevará a la pantalla de “Nuevo Cuaderno”, donde se nos piden los datos más básicos de un cuaderno de viaje. El número justo para que no se te haga pesada la creación de un cuaderno. Si se te hace pesado, puedes crear un cuaderno sin introducir ningún dato para dar agilidad a Travel Blog.

Una vez creado nuestro cuaderno de viaje, aparecerá en la lista inicial con la imagen de portada, el título y el periodo de acción correspondiente.



## Descarga

Para realizar la descarga de cuadernos, antes debemos echar un vistazo a la sección “Conexión con GeoSisTur”, donde se explica paso a paso el proceso de inicio de la comunicación con el portal web.

Teniendo ya una cuenta con GeoSisTur, en la lista de cuadernos de viaje de Travel Blog existe la opción “Descarga Cuadernos” accesible a través del menú de opciones. Solamente nos pedirán los datos de registro de nuestra cuenta GeoSisTur, y podremos acceder a los cuadernos que contiene el portal. Así podremos elegir los cuadernos a descargar, e incluso se cuenta con un buscador de cuadernos que, a través de una entrada de texto, encuentra los cuadernos de viaje cuyo título se corresponda con el texto introducido. Por ejemplo, si vamos a visitar Roma, con solo escribir Roma en el cuadro de texto aparecerán todos los cuadernos de viaje en cuyo título esté contenida la palabra Roma, y así poder disponer de una guía creada por un viajero como tú.



Tras el tiempo de descarga, dependiendo del tamaño de la cantidad de cuadernos a descargar y del tamaño de cada uno de éstos, aparecerán en nuestra lista de cuadernos de viaje para su consulta y edición, si es que el cuaderno tiene los permisos necesarios.

## Modificación

La modificación de un cuaderno puede realizarse a través del menú contextual en la lista de cuadernos de la aplicación, o a través de la opción “Modificar Cuaderno” del menú de opciones dentro de los detalles del cuaderno de viaje a modificar.



A través de ambas opciones se accede a la pantalla de modificación del cuaderno, una pantalla con los mismos campos que los que aparecen en la creación de un nuevo cuaderno, pero inicializados con la información existente del cuaderno para que puedas editarla.

**Modificar Cuaderno**

**Título**  
Lisboa

**Periodo de Acción**

Día Comienzo	Día Fin
04/12/2011	11/12/2011

**Foto de Portada**

 Elegir Portada

**Descripción del Cuaderno**

Visita de Lisboa durante el puente de la Constitucion. También visitamos otros lugares cercanos a la capital...

## Detalles de cuaderno

Una vez tenemos algún cuaderno en nuestra lista, podemos acceder a los detalles del cuaderno pulsando el ítem correspondiente en la lista. Si en vez de una pulsación simple, realizamos una pulsación continuada sobre dicho ítem, aparecerá un menú sobre la pantalla, llamado menú contextual, mostrándonos opciones que tendrán efecto sobre el cuaderno seleccionado de la lista: “Editar Cuaderno” [ver sección **Modificación**] y “Eliminar Cuaderno”, que nos eliminará dicho cuaderno previa confirmación.



### Lisboa

De 04-12-2011 a 09-12-2011

Visita de Lisboa durante el puente de la Constitucion. También visitamos otros lugares cercanos a la capital como Sintra, Estoril, Cascais y Cabo da Roca



En los detalles del cuaderno se puede ver toda la información correspondiente a dicho cuaderno, un calendario indicando los días de acción establecidos para dicho cuaderno de viaje, y una lista de entradas para cada uno de los días.

Comentar que si la descripción del cuaderno se excede en más de 3 líneas, por razones de adecuación al tamaño de la pantalla, el resto no es mostrado. En tal caso, tras una pulsación sobre el texto perteneciente a la descripción, éste expande toda la información sobre una pantalla emergente.

Ahora te toca a ti, esperamos que los detalles del cuaderno se vayan llenando de experiencias poco a poco a través de la creación de entradas de viaje. [Ver sección “**Entradas de cuadernos de viaje**”]

## Actualización

Del mismo modo que nos decargamos en Travel Blog un cuaderno creado en GeoSisTur, podemos realizar la acción opuesta, cargar un cuaderno creado en Travel Blog en GeoSisTur. Para ello disponemos de la opción “Actualizar Cuadernos” disponible en el menú de opciones. Pero antes, asegurate que la información que va a ser volcada, no elimine ninguna información anterior que quieras conservar, pues el cuaderno actualizado se sobrescribe sobre el anterior.



## Entradas de cuaderno de viaje

La imagen muestra una interfaz de usuario para un cuaderno de viaje. En la parte superior, hay un encabezado con el título 'Lisboa' y una barra de herramientas con iconos de zoom, vista y una lupa. A la izquierda, una tarjeta muestra una foto de un castillo y el título 'Lisboa' con la fecha 'De 04-12-2011 a 09-12-2011'. Debajo, un texto describe la visita. En el centro, un calendario para el 'Año 2011' muestra los días 5, 6, 7 y 8 de diciembre. Debajo del calendario, se listan las 'Entradas del 7 de Diciembre', incluyendo una entrada a las 17:15 sobre un autobús. A la derecha, una lista titulada 'Todas las entradas de Lisboa' muestra tres entradas con sus respectivos horarios. Las anotaciones con flechas rojas indican: 'Ver/ocultar todas las entradas' apunta al icono de lupa; 'Calendario' apunta al calendario; y 'Entradas del día 7' apunta a la lista de entradas del día 7.

**Ver/ocultar todas las entradas**

**Calendario**

**Entradas del día 7**

Todas las entradas de Lisboa	
10:00	Visita del palacio da Pena
13:00	Paseo por Sintra
17:15	Autobús a Cabo da Roca



Una entrada de cuaderno, es cada una de las entidades que forman un cuaderno de viaje. Se puede asemejar una entrada en Travel Blog a la visita de un lugar en la vida real. En Travel Blog, cada una de las entradas de un cuaderno serán las encargadas de almacenar los recursos multimedia y turísticos de dicho cuaderno.

Las entradas de un viaje pueden ser accedidas a través de los detalles del cuaderno correspondiente. En los detalles de dicho cuaderno, se muestran los días del período de acción en forma de hojas de calendario. Para cada una de las hojas de calendario del cuaderno viaje, existe una lista de entradas, donde por orden cronológico se sitúan las entradas perteneciente al día elegido.

En la barra de acción también se puede ver la opción “OJO” en la que se activará y desactivará el listado de todas las entradas del cuaderno de viaje independientemente del calendario.

## Creación de una entrada

Al igual que para los cuadernos de viaje, la creación de entradas de cuadernos es accesible a través la opción “MÁS” en la barra de acción o a través de la opción “Nueva Entrada” en el menú de opciones dentro del detalle de un cuaderno de viaje en concreto.



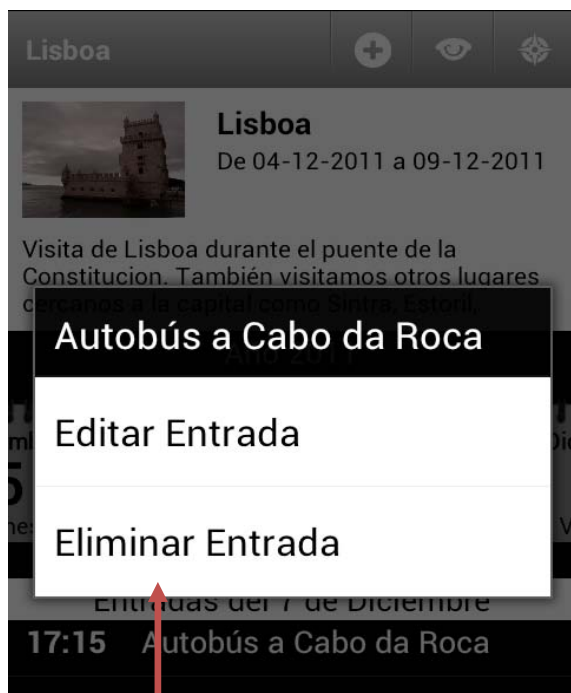
Cada una de estas dos maneras de crear una entrada nos llevará a la pantalla de creación “Nueva Entrada”, donde se nos piden los datos más básicos de una entrada de cuaderno de viaje con el fin de agilizar el proceso. Cuando confirmamos la creación de dicha entrada, tenemos la opción de guardar dicha entrada como evento de Google Calendar, utilidad creada para la planificación de viajes con antelación y para que te sirva de recordatorio.



Una vez creada nuestra entrada, aparecerá en la lista de entradas dentro del detalle del cuaderno de viaje correspondiente con la hora de inicio elegida.

## Modificación

La modificación de una entrada puede realizarse a través del menú contextual en la lista de entradas, o a través de la opción “Modificar Entrada” del menú de opciones dentro de los detalles de la entrada a modificar.



Menú contextual

This screenshot shows the 'Modificar Entrada' (Edit Entry) form. It includes the following fields and options:

- Título**: A text field containing 'Autobús a Cabo da Roca'.
- Día de la entrada**: A date picker showing '07/12/2011' (Wednesday, December 7th).
- Hora Comienzo**: A time field set to '17:15'.
- Descripción de la entrada**: A text area containing 'Ruta realizada en bus desde Sintra hasta Cabo da Roca'.
- Buttons**: 'Modificar Entrada' and 'Cancelar'.

## Detalles de Entrada



La primera vez que se acceda a los detalles de una entrada a través de la lista de entradas, aparecen los datos que has introducido durante la creación de la entrada.

Ahora es el momento de rellenar la entrada añadiendo los diferentes tipos de recursos que se pueden añadir a una entrada de viaje.

### Recursos multimedia:

A una entrada se le pueden asociar diferentes recursos multimedia a través de las opciones disponibles en el menú de opciones:

- “Añadir Foto”:

Permite la asociación de una sola fotografía a la entrada correspondiente. Esta fotografía puede tener dos orígenes: almacenada previamente en la memoria del dispositivo móvil, accesible desde diferentes aplicaciones como la Gallery de Android, o directamente desde la propia cámara de fotos del móvil.



- “Añadir Video”:

De forma análoga a añadir una fotografía, se asocian videos a una entrada.

- “Añadir Ruta”:

A través de un explorador de archivos, se asocia un archivo de extensión .gpx a la entrada correspondiente.

### Recursos turísticos:

A una entrada se le pueden asociar también diferentes recursos turísticos o POIs como monumentos, restaurantes, hoteles, eventos, etc. [ver sección **Detalle de Recurso turístico**]

- “Asociar Recurso”:

Permite la asociación de recursos turísticos disponibles en GeoSisTur previo algún tipo de filtrado como texto, cercanía, tipo, etc. Así también permite la asociación de recursos turísticos creados por el propio usuario a través del mapa. [Ver sección **Mapa**]

Estos recursos, una vez añadidos, aparecen en forma de galería, en el caso de fotos y vídeos, y en forma de listado en el caso de los recursos turísticos.

Cada uno de estos recursos puede ser accedido mediante una pulsación simple sobre cualquiera de las colecciones, ya sea en la galería como en la lista de recursos.



## Detalle de recurso turístico

Un recurso turístico o POI (Point of Interest) puede ser desde un teatro, un producto típico o una ciudad, hasta un evento, monumento, etc. Todo POI contiene información detallada de situación geográfica y contacto. Cada recurso turístico puede ser accedido a través de los detalles de la entrada a la que ha sido asociado.

Además de la información predefinida tal como teléfono, web, dirección postal, tipo, etc, el usuario puede editar dicho recurso turístico a través de la adición de fotografías propias o de comentarios valorando dicho recurso, siempre que los permisos de edición sobre el recurso lo permitan.



En los detalles de un recurso turístico, toda la información básica es accesible a través de un menú propio de opciones. Además tiene su propia galería de imágenes, de la misma manera que una entrada, y se muestran los comentarios más significativos realizados por los usuarios de la comunidad de GeoSisTur.

Por último, los comentarios pueden ser creados y añadidos por el usuario de Travel Blog, para que en todo momento se pueda tener en cuenta tu opinión.

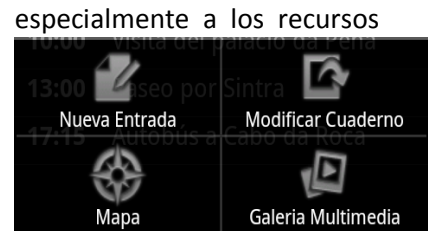
## Galería Multimedia

La galería multimedia es una zona de Travel Blog dedicada multimedia asociados a las diferentes entradas. Esta zona puede ser accedida a través de la opción “Galería Multimedia”. Esta opción es accesible desde los detalles de las tres entidades más importantes: detalles de cuaderno de viaje, detalles de entrada y detalles de recurso turístico.

Dependiendo desde que zona se acceda, la galería multimedia se rellenará de unos recursos multimedia u otros:

- Si se accede desde los detalles del cuaderno, la galería multimedia muestra toda foto y video de las entradas que forman dicho cuaderno.
- Si se accede a través de los detalles de la entrada, la galería muestra las fotos y videos asociados a dicha entrada.
- Si se accede desde los detalles de un recurso, solo las fotos y videos asociadas a éste serán mostradas en la galería.

También es posible acceder a través de las galerías de imágenes presentes en los detalles de entrada y de recurso.

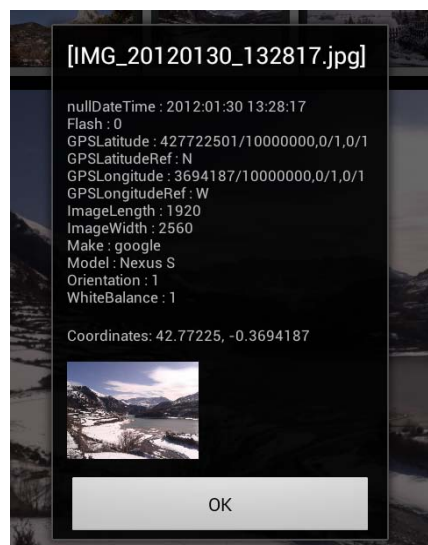


En la misma galería también se pueden realizar algunas tareas de gestión como las que se pueden ver en la imagen:

- Añadir una nueva foto a la entrada.
- Eliminar o disociar la foto central de la entrada.
- Ver la información EXIF de la imagen mostrada, solo en caso de fotografías.
- Geolocalizar en el mapa si la imagen lo permite.

Si se realiza doble click sobre la imagen mostrada, ésta se reproduce según la aplicación Android nativa compatible con dicho fichero. Por ejemplo, si se trata de un vídeo, al hacer doble click es reproducido por la aplicación nativa de vídeo.

Por último comentar que la galería permite el paso de fotografías por deslizamiento horizontal, zoom mediante pellizco y navegación sobre la imagen mostrada.

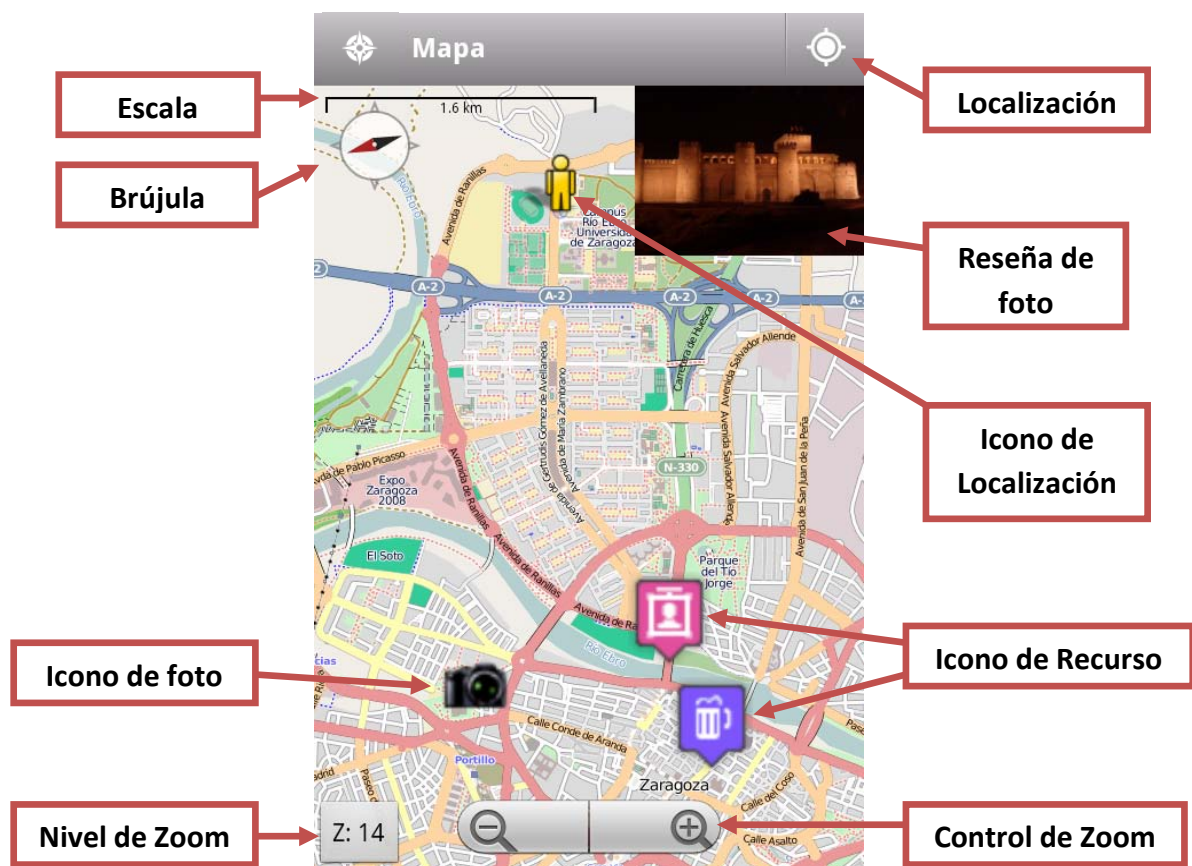


## Mapa

El mapa es una de las partes principales de la aplicación debido a su utilidad de cara al viajero. Es accesible tanto en la opción “BRÚJULA” presente en la barra de acción de las pantallas principales, así como en la opción “Mapa” que podemos encontrar en el menú de opciones de las pantallas principales.



En la pantalla del mapa nos encontraremos con una barra de escala, una brújula y el nivel de zoom actual para ayudar a la mejor comprensión de la zona a explorar.



En la barra de acción existe una opción también accesible desde el menú de opciones que es la de localización. Permite al usuario hallar la localización donde se encuentra en ese mismo momento. La localización se basa en red de datos, o en GPS. Si queremos que se realice con éxito, al menos una de las dos opciones debe estar activada.

El aumento y decremento del nivel de zoom se puede realizar de dos maneras:

- Mediante los controles que aparecen cuando el mapa es tocado.
- A través la pantalla multitáctil a través del gesto “pellizco”.

Haciendo doble click sobre una localización se produce un aumento del zoom además del centrado del mapa sobre la zona pulsada.

## Descargar Mapa

Permite la descarga del mapa que se visualiza en pantalla para que cuando no se disponga de conexión a Internet, el mapa pueda estar disponible.

Se accede a través del menú de opciones de la pantalla que muestra el mapa.

La descarga de mapas se debe realizar cuando existe una conexión de datos.

Permite la descarga hasta un nivel indicado de zoom. Es decir, si para un mapa descargas el nivel 14, quieres decir que más allá del nivel 14 no podrás hacer zoom. Un ejemplo gráfico lo puedes ver en la parte inferior de esta página.

Cada una de las imágenes descargadas de la zona del mapa en cuestión se almacenan en la memoria del dispositivo móvil, por lo que se debe tener cuidado sobre la memoria disponible que dispongas porque puede ocurrir un error en la descarga debido a la falta de espacio en memoria.

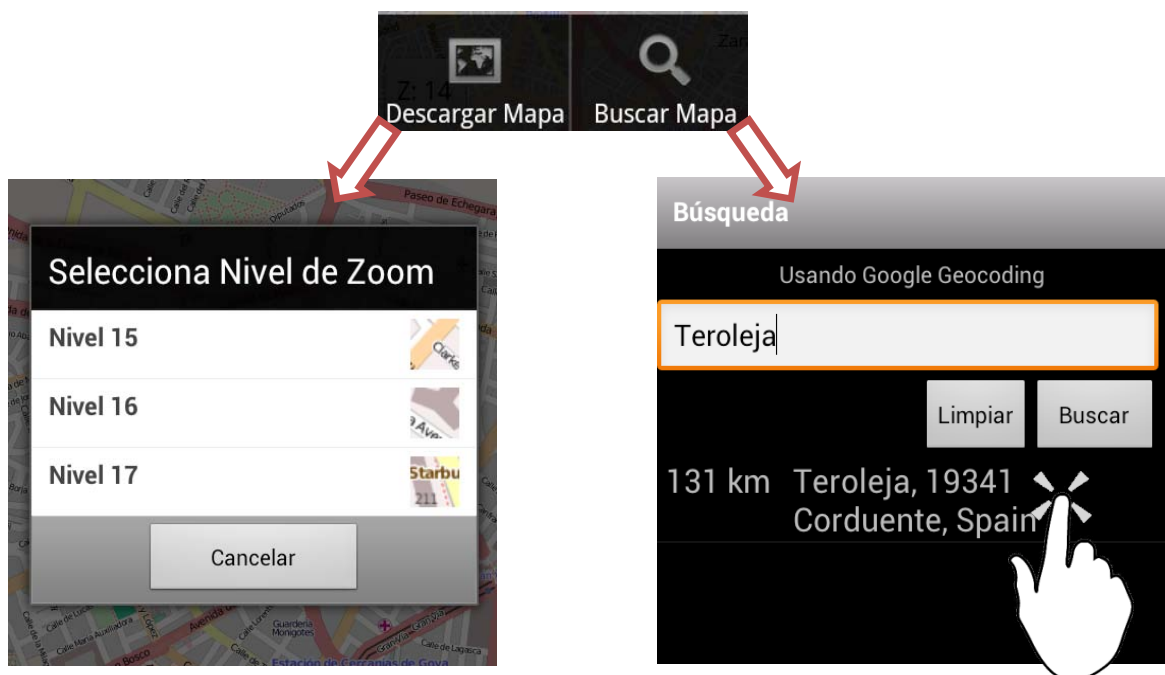
Cuando el usuario se encuentre en una zona sin acceso a datos, el mapa descargado previamente cuando se disponía de conexión, será automáticamente visible.

## Búsqueda en el Mapa

A través de una entrada de texto, busca localizaciones en el mapa que corresponda con la entrada. El resultado que arroja la búsqueda es un listado de lugares junto con la distancia a la que se encuentran respecto de la posición actual.

Para acceder a la posición de uno de los resultados, tan sólo se debe hacer click sobre el ítem correspondiente en la lista.

A la opción de búsqueda en el mapa se accede a través del menú de opciones de la pantalla que muestra el mapa.



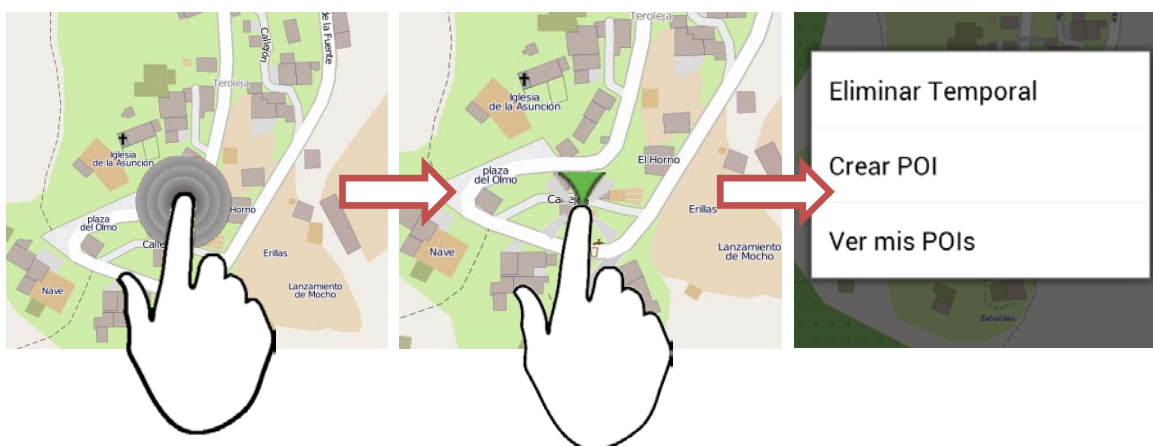


## Creación de POI

Si se mantiene pulsado el mapa en una posición fija, el mapa se centra sobre la zona pulsada además de aparecer un icono temporal. Si lo pulsamos nos aparecerá una serie de opciones:

- Eliminar temporal: borra el icono creado en la pulsación.
- Crear POI: permite la creación de un POI personal tras la introducción de los datos pertinentes.
- Ver mis POIs: muestra todos los POIs creados por el usuario en forma de listado.

Podemos ver la secuencia descrita antes en la imagen que aparece a continuación.



La creación de un POI personal requiere de un formulario como el siguiente. Solo debes rellenar al menos el nombre y el tipo y el POI creado estará disponible a tu disposición para su asociación a una entrada.

**Nuevo Poi**

Nombre:  
Nombre del recurso

Dirección:  
Dirección postal del lugar

Teléfono:  
Teléfono de contacto

Website:  
Sitio web del lugar

Tipo:  
Indefinido Elegir tipo

Lat: 38.7855 Lon: -9.390134

Crear POI Cancelar

**Escoge Tipo de Recurso**

>Recursos>Evento

	Actividad	<span>+</span>	<input type="checkbox"/>
	Espectáculo público	<span>+</span>	<input type="checkbox"/>
	Evento cultural		<input type="checkbox"/>
	Evento deportivo		<input type="checkbox"/>
	Feria		<input type="checkbox"/>
	Festival	<span>+</span>	<input type="checkbox"/>
	Mercado		<input type="checkbox"/>

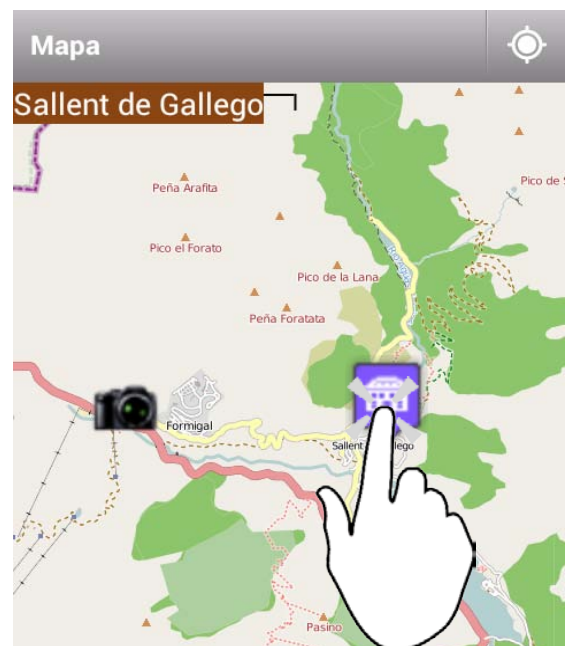
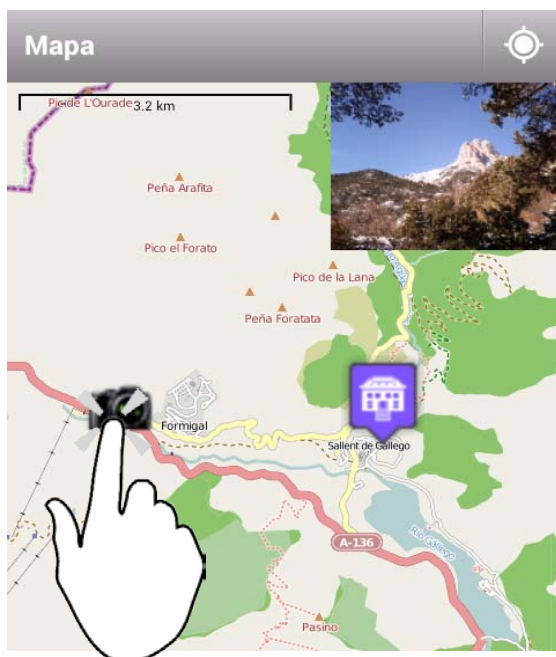
El mapa nos muestra información geolocalizada automáticamente dependiendo del lugar desde donde sea accedido:

- Desde el listado de cuadernos: el mapa no geolocaliza recurso alguno.
- Desde los detalles de un cuaderno: el mapa geolocaliza todas aquellas fotos y recursos turísticos que estén contenidos dentro de las entradas de dicho cuaderno.
- Desde los detalles de una entrada: el mapa geolocaliza todas aquellas fotos y recursos turísticos que estén contenidos dentro de la entrada.
- Desde la galería multimedia (solo aparece la opción “Ver en Mapa” si el multimedia dispone de geolocalización): el mapa geolocaliza la imagen seleccionada.
- Desde los detalles de un recurso turístico: el acceso al mapa geolocaliza el recurso dado.

Si los iconos sobre el mapa son recursos turísticos, estos estarán representados mediante una imagen que denota el tipo de recurso.

En el caso de que los iconos sobre el mapa sean fotografías, un icono representando una cámara de fotos denotará su posición.

Si se realiza una pulsación sobre cualquiera de estos dos tipos de iconos, se representará una breve reseña del icono pulsado: el nombre en caso de que el icono pulsado sea un recurso turístico, y la imagen en tamaño reducido en el caso de que el icono sea una cámara de fotos.





## Conexión con GeoSisTur

Como ya hemos comentado, Travel Blog mantiene una conexión con la plataforma web turística GeoSisTur. A través de esta plataforma podremos obtener numerosa información como cuadernos de viaje, recursos turísticos establecidos, rutas, etc.

Para acceder a esta información disponible en Internet, antes deberemos registrarnos en el portal web.

Una vez realizado el registro, cada vez que se desee acceder a la información de GeoSisTur a través de las acciones enumeradas más abajo, se nos pedirán los datos de registro para poder disponer de ella. Para evitar la repetición de tareas y pérdida de tiempo, únicamente introduciendo los datos de registro la primera vez correctamente, éstos se almacenan en las preferencias de la aplicación, accesibles en caso de modificación.

Por supuesto, para la correcta conexión con el portal web debemos disponer de red de datos, ya sea WIFI o móvil.

Las diferentes tareas en las que necesitaremos conexión con GeoSisTur son:

- Registro de usuario.
- Descarga de cuadernos.
- Actualización de cuadernos.
- Búsqueda de cuadernos.
- Búsqueda de recursos turísticos.

A screenshot of a login dialog box titled "Geosistur". It features two input fields: "Usuario:" with the text "sergiomr" and "Password:" with seven dots. Below the fields are two buttons: "Aceptar" and "Cancelar". The dialog box has a dark background and a light border.

## Pantallas de Travel Blog

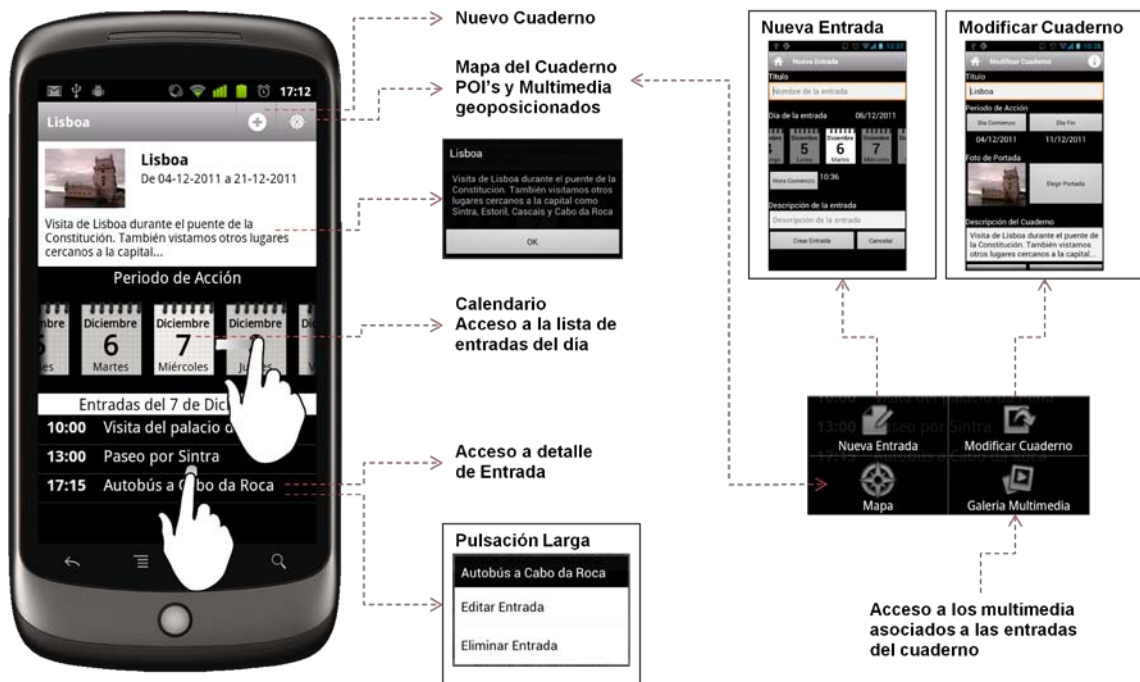
### Pantalla de bienvenida



### Lista de cuadernos de viaje



## Detalle de cuaderno de viaje



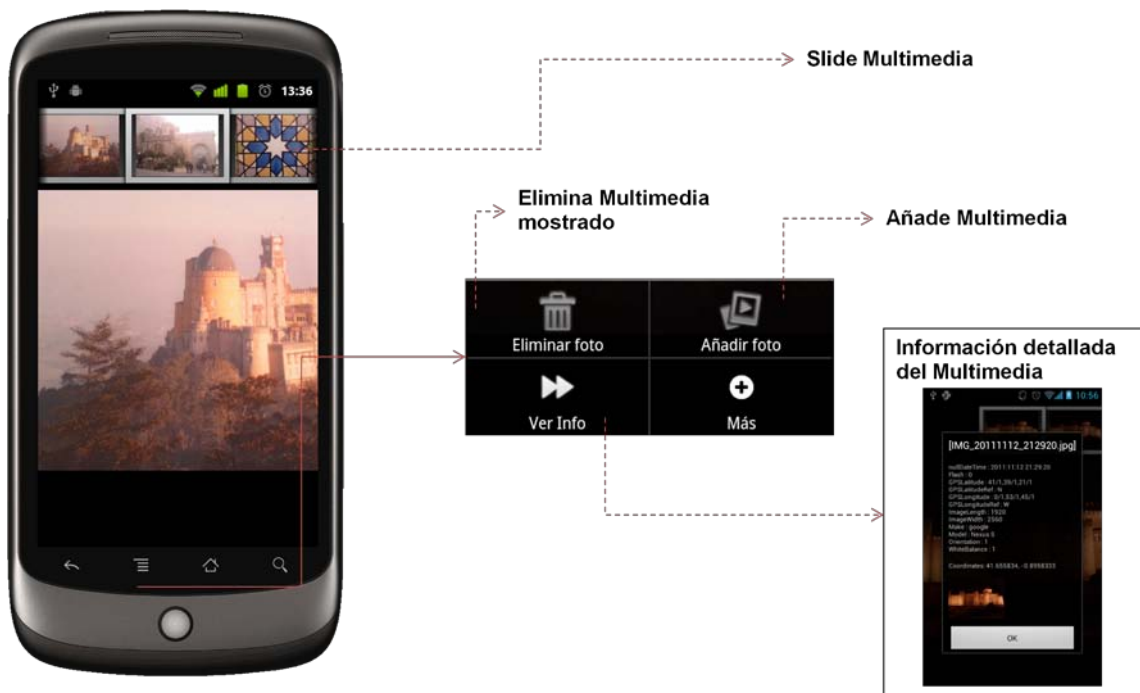
## Detalle de entrada



## Detalle de recurso turístico



## Galería Multimedia



## Mapa



## Parte III

### Acrónimos, Figuras, Bibliografía

# Acrónimos

**API** Application Programming Interface

**DGA** Diputación General de Aragón

**GPS** Global Positioning System

**HTTP** HyperText Transfer Protocol

**I+D+i** Investigación, desarrollo e innovación

**JSON** JavaScript Object Notation

**NFC** Near Field Communication

**ORM** Object-Relational Mapping

**OSM** Open Street Maps

**PFC** Proyecto Fin de Carrera

**POI** Point Of Interest

**REST** Representational State Transfer

**SD** Secure Digital

**SDK** Software Development Kit

**SOAP** Simple Object Access Protocol

**SQL** Structured Query Language

**WSDL** Web Services Description Language

**XML** eXtensible Markup Language

# Índice de figuras

1.1. Reparto de las versiones de Android a fecha de 3 de enero de 2012 . . .	4
2.1. Arquitectura del sistema de la aplicación Travel Blog . . . . .	16
2.2. Modelo de datos de la aplicación móvil Travel Blog . . . . .	17
2.3. Ventanas de la aplicación realizadas durante la fase de diseño . . . . .	18
2.4. Editor XML del entorno de programación utilizado para el diseño de la interfaz de usuario de TravelBlog . . . . .	20
2.5. Arquitectura de ORMLite . . . . .	21
2.6. Acciones gestuales de interacción con Travel Blog: (a)-Pulsación simple; b()-Desplazamiento horizontal; c()-Desplazamiento vertical; d()-Aumento de nivel de zoom . . . . .	25
2.7. Calendario temporal inicial de Travel Blog . . . . .	29
2.8. Calendario creado para la gestión de las entradas de un cuaderno . . .	30
2.9. Visor de mapas modificado para Travel Blog . . . . .	32
2.10. Arquitectura de la comunicación de Travel Blog con GeoSisTur . . . .	34
A.1. Cronograma final del PFC. . . . .	41
C.1. Prototipos realizados con Pencil en la fase de diseño . . . . .	47
C.2. Navegación de ventanas entre las principales pantallas diseñadas . . .	48
C.3. Modelo de datos de la aplicación móvil Travel Blog . . . . .	49
C.4. Detalle de los atributos de las entidades más importantes del modelo de datos . . . . .	49
D.1. Arquitectura del sistema operativo Android . . . . .	50
D.2. Arquitectura de la aplicación Travel Blog . . . . .	52



# Bibliografía

- [1] Plataforma GeoSisTur <http://www.geosistur.com/> 1.2.4
- [2] Mark L. Murphy: *The Busy Coders Guide to Android Development*, v3.6. CommonsWare , 2011  
ISBN: 978-0-9816780-0-9
- [3] Mark L. Murphy: *The Busy Coder's Guide to Advanced Android Development*, v1.9. CommonsWare, 2010  
[http://commonsware.com/AdvAndroid/AdvAndroid-1\\_9-CC.pdf](http://commonsware.com/AdvAndroid/AdvAndroid-1_9-CC.pdf)  
ISBN: 978-0-9816780-1-6
- [4] Grant Allen and Mike Owens: *The Definitive Guide to SQLite*, Segunda edición. Apress, 2010  
ISBN: 978-1-4302-3225-4
- [5] Gray Watson: *ORMLite Package*, v4.25. 2011  
<http://ormlite.com/docs/ormlite.pdf>
- [6] Android Developers  
<http://developer.android.com/index.html>
- [7] Salvador Gómez: *Tutorial Android por Sgoliver*  
<http://www.sgoliver.net/blog/?p=1313>
- [8] Preguntas y respuestas sobre programación  
<http://stackoverflow.com>
- [9] Cyril Mottier: *Librería GreenDroid*  
<http://android.cyrilmottier.com/?p=240>
- [10] OSMDroid - OpenStreetMaps-Tools for Android  
<http://code.google.com/p/osmdroid/>
- [11] Maps-Minus: OpenStreetMaps.org offline map viewer  
<http://code.google.com/p/maps-minus/> 2.4.2
- [12] OsmAnd: *Navigation & routing based on Open Street Maps for Android devices*  
<http://code.google.com/p/osmand/>

- [13] JavaCodeGeeks: *Android JSON Parsing with Gson Tutorial*  
<http://www.javacodegeeks.com/2011/01/android-json-parsing-gson-tutorial.html>

**Nota:** Las direcciones de Internet que aparecen en esta bibliografía están revisadas y accedidas a fecha 09/02/2012.